




PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on September 3, 2004.


Alex Martinez

Applicant : Shougo Imada, et al.
Application No. : 10/769,488
Filed : January 29, 2004
Title : MICROCOMPUTER LOGIC DEVELOPMENT

Grp./Div. : 2825
Examiner : N/A

Docket No. : 51879/DBP/A400

**LETTER FORWARDING CERTIFIED
PRIORITY DOCUMENT**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

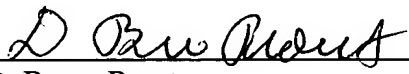
Post Office Box 7068
Pasadena, CA 91109-7068
September 3, 2004

Commissioner:

Enclosed is a certified copy of Japanese Patent Application No. 2003-024733, which was filed on January 31, 2003, the priority of which is claimed in the above-identified application.

Respectfully submitted,

CHRISTIE, PARKER & HALE, LLP

By 
D. Bruce Prout
Reg. No. 20,958
626/795-9900

DBP/aam
Enclosure: Certified copy of patent application

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出 願 年 月 日
Date of Application: 2 0 0 3 年 1 月 3 1 日

出 願 番 号
Application Number: 特 願 2 0 0 3 - 0 2 4 7 3 3
[J P 2 0 0 3 - 0 2 4 7 3 3]
ST. 10/C] :

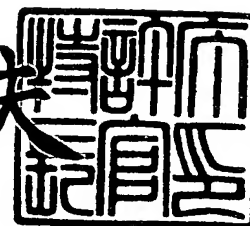
願 人
Applicant(s): 富士通テン株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2 0 0 4 年 1 月 2 1 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 1025213

【提出日】 平成15年 1月31日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06F 11/22
G06F 11/25

【発明の名称】 マイクロコンピュータのロジック開発装置

【請求項の数】 21

【発明者】

 【住所又は居所】 兵庫県神戸市兵庫区御所通 1 丁目 2 番 2 8 号 富士通テ
 ン株式会社内

 【氏名】 今田 昭吾

【発明者】

 【住所又は居所】 兵庫県神戸市兵庫区御所通 1 丁目 2 番 2 8 号 富士通テ
 ン株式会社内

 【氏名】 柏原 俊浩

【発明者】

 【住所又は居所】 兵庫県神戸市兵庫区御所通 1 丁目 2 番 2 8 号 富士通テ
 ン株式会社内

 【氏名】 樋口 崇

【特許出願人】

 【識別番号】 000237592

 【氏名又は名称】 富士通テン株式会社

【代理人】

 【識別番号】 100077517

 【弁理士】

 【氏名又は名称】 石田 敬

 【電話番号】 03-5470-1900

【選任した代理人】

【識別番号】 100092624

【弁理士】

【氏名又は名称】 鶴田 準一

【選任した代理人】

【識別番号】 100100871

【弁理士】

【氏名又は名称】 土屋 繁

【選任した代理人】

【識別番号】 100082898

【弁理士】

【氏名又は名称】 西山 雅也

【選任した代理人】

【識別番号】 100081330

【弁理士】

【氏名又は名称】 樋口 外治

【手数料の表示】

【予納台帳番号】 036135

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9814498

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 マイクロコンピュータのロジック開発装置

【特許請求の範囲】

【請求項 1】 電子制御ユニットに組み込まれて使用される組込み用マイクロコンピュータのロジック開発装置であって、

アプリケーション処理機能と通信機能を備えた中央ブロックと、

マイクロコンピュータの周辺装置を擬似的にソフトウェアで実現して入出力処理を行う疑似マイクロコンピュータ周辺装置、演算機能、及び通信機能を有し、前記中央ブロックと P C I バスで接続される周辺ブロックと、

電子制御ユニットのハードウェアに相当する回路を有し、前記周辺ブロックに接続するインタフェース回路ブロックとを備え、

前記中央ブロックの通信機能と、前記周辺ブロックの前記疑似マイクロコンピュータ周辺装置とが前記 P C I バスで接続され、この P C I バスを介して前記通信機能と前記疑似マイクロコンピュータ周辺装置との間で、直接データの送受信が行えるように構成されていることを特徴とするマイクロコンピュータのロジック開発装置。

【請求項 2】 電子制御ユニットに組み込まれて使用される組込み用マイクロコンピュータのロジック開発装置であって、

アプリケーション処理機能と通信機能を備えた中央ブロックと、

マイクロコンピュータの周辺装置を擬似的にソフトウェアで実現して入出力処理を行う疑似マイクロコンピュータ周辺装置、演算機能、及び通信機能を有し、前記中央ブロックと P C I バスで接続される周辺ブロックと、

電子制御ユニットのハードウェアに相当する回路を有し、前記周辺ブロックに接続するインタフェース回路ブロックとを備え、

前記中央ブロックの通信機能と、前記疑似マイクロコンピュータ周辺装置との間にある前記演算機能に、バスコントローラが設けられ、前記 P C I バスで前記中央ブロックの通信機能とこのバスコントローラとの間が接続され、前記バスコントローラと前記疑似マイクロコンピュータ周辺装置との間は内部バスで接続され、前記 P C I バス、前記バスコントローラ、及び前記内部バスを介して、前記

通信機能と前記疑似マイクロコンピュータ周辺装置との間で、直接データの送受信が行われるように構成したことを特徴とするマイクロコンピュータのロジック開発装置。

【請求項 3】 前記中央ブロックの通信機能と前記 P C I バスとの間に仮想入出力レジスタが設けられており、

データの入出力操作のタイミングにおいて、この仮想入出力レジスタに送受信データが一旦蓄積されることにより、この仮想入出力レジスタが実際のマイクロコンピュータの入出力レジスタと同様の振る舞いをするように構成されていることを特徴とする請求項 1 又は 2 に記載のマイクロコンピュータのロジック開発装置。

【請求項 4】 前記アプリケーション処理機能の処理対象が車両であり、前記ロジック開発装置にイグニッションスイッチが設けられており、前記ロジック開発装置がこのイグニッションスイッチのオン／オフに連動して、実際の電子制御ユニット同様に前記車両の制御ソフトウェア処理の開始／停止が実行されることを特徴とする請求項 1 から 3 の何れか 1 項に記載のマイクロコンピュータのロジック開発装置。

【請求項 5】 前記インタフェース回路ブロックに備えられた前記電子制御ユニットのハードウェアに相当する回路に、マイクロコンピュータを内蔵した機能回路が少なくとも 1 つ存在し、この機能回路は、前記イグニッションスイッチのオンでは動作せず、前記中央ブロックの処理開始タイミングに同期してオンするように構成されていることを特徴とする請求項 4 に記載のマイクロコンピュータのロジック開発装置。

【請求項 6】 前記機能回路には、前記イグニッションスイッチのオンで動作する電源回路と、この電源回路からの信号と、前記中央ブロックからの信号が共にアクティブになった時に前記マイクロコンピュータを動作させる論理回路が設けられていることを特徴とする請求項 5 に記載のマイクロコンピュータのロジック開発装置。

【請求項 7】 前記イグニッションスイッチのオフ時に、保持すべきデータの値が、前記ロジック開発装置に接続されている外部記憶装置内のメモリと前記

ロジック開発装置内のメモリの何れかに保存され、前記イグニッションスイッチのオン時に、前記メモリから前記保持すべきデータの値が読み出されて復元され、前記ロジック開発装置にバックアップメモリが備えられているのと同等の機能が実現されることを特徴とする請求項 4 から 6 の何れか 1 項に記載のマイクロコンピュータのロジック開発装置。

【請求項 8】 前記ロジック開発装置の電源がオンされた後に、前記イグニッションスイッチがオンされるまでの間に、前記中央ブロックの初期化ルーチンにより、ポートの初期値が設定されることを特徴とする請求項 4 から 7 の何れか 1 項に記載のマイクロコンピュータのロジック開発装置。

【請求項 9】 前記 P C I バスに、前記周辺ブロックから前記中央ブロックへの割込み要求を発行することができる 1 系統の割込み信号ラインが設けられており、前記中央ブロックの前記アプリケーション処理機能は、前記周辺ブロックによって前記割込み信号ラインがアクティブになった時に割込み処理を受け付けるようになっていると共に、割込み処理の受け付け終了後に、前記割込み信号ラインがノンアクティブになるように動作することを特徴とする請求項 1 から 8 の何れか 1 項に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 0】 前記中央ブロックの前記アプリケーション処理機能は、前記割込み処理の終了時に、前記割込み信号ラインがノンアクティブであることを確認するように動作することを特徴とする請求項 9 に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 1】 前記中央ブロックの前記アプリケーション処理機能は、前記割込み処理の終了時に前記割込み信号ラインがアクティブとなっている時は、前記割込み信号ラインをノンアクティブに戻すように動作することを特徴とする請求項 1 0 に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 2】 前記周辺ブロックの演算機能にデータの一時取り込み機能が設けられており、前記中央ブロックと、前記周辺ブロックの前記疑似マイクロコンピュータ周辺装置との間で大量のデータが送受信される場合に、この大量のデータは前記中央ブロックとこの演算機能との間でバースト転送により送受信され、前記演算機能と前記疑似マイクロコンピュータ周辺装置との間には非バースト

転送により少量ずつデータの送受信が行われることを特徴とする請求項 1 から 1 1 の何れか 1 項に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 3】 前記中央ブロックの前記アプリケーション処理機能により、前記割込み処理の受け付け終了後に前記 P C I バスを通じて前記疑似マイクロコンピュータ周辺装置から割込みフラグが取得され、前記アプリケーション処理機能によって割込みフラグが取得されると、前記疑似マイクロコンピュータ周辺装置にある当該割込みフラグは、前記アプリケーション処理機能によりクリアされることを特徴とする請求項 9 に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 4】 前記中央ブロックの前記アプリケーション処理機能により割込みフラグが取得されると、前記アプリケーション処理機能により、取得した割込みフラグに付随する処理が実行されることを特徴とする請求項 1 3 に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 5】 前記中央ブロックの前記アプリケーション処理機能により、前記割込み処理の受け付け終了後に前記 P C I バスを通じて前記疑似マイクロコンピュータ周辺装置から複数の割込みフラグが前記アプリケーション処理機能に取得された場合に、

前記アプリケーション処理機能によって優先度の高い割込みフラグが 1 つ選択されてそのフラグに対する処理が実行され、処理終了後に、前記疑似マイクロコンピュータ周辺装置にある当該処理完了割込みフラグは、前記アプリケーション処理機能によりクリアされることを特徴とする請求項 9 に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 6】 前記アプリケーション処理機能によって優先度の高い割込みフラグが 1 つ選択されてそのフラグに対する処理が実行された後は、前記アプリケーション処理機能により、再度前記 P C I バスを通じて前記疑似マイクロコンピュータ周辺装置から複数の割込みフラグが前記アプリケーション処理機能に取得されることを特徴とする請求項 1 5 に記載のマイクロコンピュータのロジック開発装置。

【請求項 1 7】 前記割込みフラグは、前記疑似マイクロコンピュータ周辺

装置の 1 つのレジスタの連続するアドレスに、まとめて記憶されていることを特徴とする請求項 13 から 16 の何れか 1 項に記載のマイクロコンピュータのロジック開発装置。

【請求項 18】 前記周辺ブロックが複数個設けられており、

各周辺ブロックに設けられた各レジスタには、各周辺ブロックにおいて割込み処理を発生させる各リソースの割込みフラグがそれぞれ格納されており、

第 1 番目の周辺ブロックに設けられた前記レジスタには、第 1 番目の周辺ブロックにおける各リソースの割込みフラグがそれぞれ格納されていると共に、残りの周辺ブロックにおける各リソースの割込みフラグの有無を示す拡張割込みフラグが、各周辺ブロック毎に格納されていることを特徴とする請求項 17 に記載のマイクロコンピュータのロジック開発装置。

【請求項 19】 前記拡張割込みフラグにより、残りの何れかの周辺ブロックの前記レジスタに割込みフラグが格納されていることが判明した場合には、前記アプリケーション処理機能により、前記残りの周辺ブロックの当該レジスタから前記割込みフラグが取得されることを特徴とする請求項 18 に記載のマイクロコンピュータのロジック開発装置。

【請求項 20】 前記周辺ブロックが複数個設けられており、

第 1 番目の周辺ブロックのみにフリーランタイムが存在し、

前記第 1 番目の周辺ブロックには、少なくとも前記フリーランタイムのタイム値に同期して動作するリソースが配置され、

残りの周辺ブロックには、前記フリーランタイムに依存しないリソースのみが配置されていることを特徴とする請求項 1 から 19 の何れか 1 項に記載のマイクロコンピュータのロジック開発装置。

【請求項 21】 前記フリーランタイムのタイム値に同期して動作するリソースがコンペア、キャプチャであり、

前記フリーランタイムに依存しないリソースが PWM、通信、AD、及びポートであることを特徴とする請求項 20 に記載のマイクロコンピュータのロジック開発装置。

【発明の詳細な説明】

【0001】**【発明の属する技術分野】**

本発明はマイクロコンピュータのロジック開発装置に関し、特に、電子制御機器に組み込まれて使用されている組込み用マイクロコンピュータにおけるロジックを開発するための装置に関する。

【0002】**【従来の技術】**

従来、電子制御機器の制御、例えば、ECU（電子制御ユニット）によって制御されるエンジンの制御は、排気ガス規制等の法規要求、及びマイクロコンピュータの性能向上に応じた性能向上を図るため、年々改良を加える必要があり、現状のECUの性能に先行して新しいロジックが開発されているのが現状である。このため、先行ロジックは、性能向上が見込まれる次期マイクロコンピュータがターゲットとなることが多い。また、先行ロジックに必要とされる性能から性能の向上した次期マイクロコンピュータの選定が行われる。

【0003】

しかしながら、性能の向上した次期マイクロコンピュータを組み込んだECUは実在しないため、前述の先行ロジックは、多くの場合は現状のマイクロコンピュータをベースにして開発が行われる。

【0004】

ところが、現状のマイクロコンピュータを組み込んだECUを用いて先行ロジックの開発を行う場合には、以下のような問題点があった。

- (1) マイクロコンピュータのCPUの処理能力が不足している。
- (2) マイクロコンピュータのメモリの容量が足りない。
- (3) 周辺リソースが足りない。
- (4) 次期ECUを開発して製作する迄に時間がかかる。

【0005】

そして、このような問題点の存在により、次期マイクロコンピュータの開発が遅れ、この次期マイクロコンピュータを組み込んだECUに制御される電子制御機器の新製品の開発に支障をきたしていた。

【0 0 0 6】

なお、現状の電子制御機器に組み込まれる組み込み用マイクロコンピュータは、電子制御機器のコストを低く抑えるために、現状のシステムに最適な仕様でCPU性能、及び周辺機能が選択されており、また、CPUやマイクロコンピュータ周辺リソースが1つのパッケージの中に組み込まれているため、それぞれの機能変更はマイクロコンピュータを変更しない限り不可能である。また、組み込み用マイクロコンピュータのロジックの開発を行うためには、CPU機能に対しては先行ロジックを処理するのに必要な処理性能が必要であり、更に、マイクロコンピュータ周辺リソースには先行システムに合わせたリソースを確保する必要がある。その上、新たなマイクロコンピュータの開発を行う度に、マイクロコンピュータに合わせてECUを製作する必要がある。

【0 0 0 7】

この課題に対して、汎用的なCPU、メモリ（RAM、ROM）、プログラマブル論理回路（FPGA）、及びこれらを接続するプログラマブル配線素子（FPIC）をプリント基板上に交換可能に設けて1チップマイコン用検証ボードを作成し、組み込み用マイクロコンピュータの機能をこの1チップマイコン用検証ボードによって実現しているものがある。（例えば、特許文献1参照）。

【0 0 0 8】

【特許文献1】

特開平8-16425号公報（図1、図2）

この1チップマイコン用検証ボードでは、1チップマイクロコンピュータを適用するシステムに対応するユーザ回路、I/O部及びレジスタ部に対応する論理機能をFPGAに書き込んでおき、FPICを用いてCPU、メモリ、及びFPGA間の入出力端子の接続状態を設定している。また、異なる1チップマイコン用検証ボードを開発する場合には、適用するシステムに対応したROM、FPGA、及びFPICを新たに作り、対応するソケットのデバイスと交換することによってこれを行っていた。

【0 0 0 9】

これに対して、本発明者らは、このような先行ロジックの開発に際して汎用性

を向上させるために、現状の電子制御機器に組み込まれる組込み用マイクロコンピュータを、外部に設けた高性能のマイクロコンピュータに置き換えることにより、先行ロジックを開発することができるマイクロコンピュータのロジック開発装置を既に提案した（特願 2 0 0 1 - 3 6 7 4 9 6 号）。

【0 0 1 0】

このマイクロコンピュータのロジック開発装置では、E C U は一般に、マイクロコンピュータの C P U 機能を備えたマザーボード、マイクロコンピュータの入出力（I / O）のリソース機能を備えたコアボード、及び、ハードウェア機能を備えた I F ボードから構成されている。マザーボードとコアボード間は P C I バスで接続され、I / O 情報の通信が行われる。各ボード自体の性能は、各ボードに搭載される部品の性能によって決定される。

【0 0 1 1】

【発明が解決しようとする課題】

しかしながら、各ボードの性能をマイクロコンピュータのロジック開発装置自体の性能として有効に引き出せるかどうかは、マザーボードとコアボードの間における I / O 情報の通信速度、マザーボードとコアボードにおける演算処理能力が大きく影響することが知られている。例えば、電子制御機器が高性能のエンジンの制御システムである場合には、以下のような課題が発生する。

- （１） I / O データの異常によるエンジン制御システムへの影響が大である。
- （２） 厳しいタイミングでの I / O 操作を行う際の処理が増加し、処理時間が増大する。
- （３） データ処理量の多い演算処理が増加する。

【0 0 1 2】

そこで、本発明の目的は、マザーボードとコアボード間における I / O 情報の確実な通信を実現できると共に、I / O 情報通信のスピードを向上させること、及び、マザーボードとコアボードの演算処理能力の向上を図ることができて、より高性能のエンジン制御システムに対応することが可能なマイクロコンピュータのロジック開発装置を提供することである。

【0 0 1 3】

【課題を解決するための手段】

前記目的を達成する本発明のマイクロコンピュータのロジック開発装置は以下の第 1 から第 2 1 の形態を取り得ることを特徴としている。

【0 0 1 4】

第 1 の形態は、制御ユニットに組み込まれて使用される組込み用マイクロコンピュータのロジック開発装置であって、アプリケーション処理機能と通信機能を備えた中央ブロックと、マイクロコンピュータの周辺装置を擬似的にソフトウェアで実現して入出力処理を行う疑似マイクロコンピュータ周辺装置、演算機能、及び通信機能を有し、中央ブロックと P C I バスで接続される周辺ブロックと、電子制御ユニットのハードウェアに相当する回路を備え、周辺ブロックに接続するインタフェース回路ブロックとを備え、中央ブロックの通信機能と、周辺ブロックの疑似マイクロコンピュータ周辺装置とが P C I バスで接続され、この P C I バスを介して通信機能と疑似マイクロコンピュータ周辺装置との間で、直接データの送受信を行うものである。

【0 0 1 5】

第 2 の形態は、制御ユニットに組み込まれて使用される組込み用マイクロコンピュータのロジック開発装置であって、アプリケーション処理機能と通信機能を備えた中央ブロックと、マイクロコンピュータの周辺装置を擬似的にソフトウェアで実現して入出力処理を行う疑似マイクロコンピュータ周辺装置、演算機能、及び通信機能を有し、中央ブロックと P C I バスで接続される周辺ブロックと、電子制御ユニットのハードウェアに相当する回路を備え、周辺ブロックに接続するインタフェース回路ブロックとを備え、中央ブロックの通信機能と、疑似マイクロコンピュータ周辺装置との間にある演算機能に、バスコントローラが設けられ、P C I バスで中央ブロックの通信機能とこのバスコントローラとの間が接続され、バスコントローラと疑似マイクロコンピュータ周辺装置との間は内部バスで接続され、P C I バス、バスコントローラ、及び内部バスを介して、通信機能と疑似マイクロコンピュータ周辺装置との間で、直接データの送受信を行うものである。

【0 0 1 6】

第3の形態は、第1又は第2の形態において、中央ブロックの通信機能とP C Iバスとの間に仮想入出力レジスタが設けられており、データの入出力操作のタイミングにおいて、この仮想入出力レジスタに送受信データが一旦蓄積されることにより、この仮想入出力レジスタが実際のマイクロコンピュータの入出力レジスタと同様の振る舞いをするように構成されているものである。

【0017】

第4の形態では、第1から第3の形態の何れかのアプリケーション処理機能の処理対象が車両であり、ロジック開発装置にイグニッションスイッチが設けられており、ロジック開発装置がこのイグニッションスイッチのオン／オフに連動して、実際のECU1同様に車両の制御ソフトウェア処理の開始／停止が実行されるものである。

【0018】

第5の形態は、第4の形態のインタフェース回路ブロックに備えられた電子制御ユニットのハードウェアに相当する回路に、マイクロコンピュータを内蔵した機能回路が少なくとも1つ存在し、この機能回路は、イグニッションスイッチのオンでは動作せず、中央ブロックの処理開始タイミングに同期してオンするように構成されているものである。

【0019】

第6の形態は、第5の形態の機能回路に、イグニッションスイッチのオンで動作する電源回路と、この電源回路からの信号と、中央ブロックからの信号が共にアクティブになった時にマイクロコンピュータを動作させる論理回路が設けられているものである。

【0020】

第7の形態では、第4から第6の形態のイグニッションスイッチのオフ時に、保持すべきデータの値が、ロジック開発装置に接続されている外部記憶装置内のメモリとロジック開発装置内のメモリの何れかに保存され、イグニッションスイッチのオン時に、外部記憶装置から保持すべきデータの値が読み出されて復元され、ロジック開発装置にバックアップメモリが備えられているのと同等の機能が実現されるものである。

【0021】

第8の形態は、第4から7の形態のロジック開発装置の電源がオンされた後に、イグニッションスイッチがオンされるまでの間に、中央ブロックの初期化ルーチンにより、ポートの初期値が設定されるものである。

【0022】

第9の形態は、第1から8の何れかの形態のP C Iバスに、周辺ブロックから中央ブロックへの割込み要求を発行することができる1系統の割込み信号ラインが設けられており、中央ブロックのアプリケーション処理機能は、周辺ブロックによって割込み信号ラインがアクティブになった時に割込み処理を受け付けるようになっていると共に、割込み処理の受け付け終了後に、前記割込み信号ラインがノンアクティブになるように動作することを特徴とするものである。

【0023】

第10の形態では、第9の形態の中央ブロックのアプリケーション処理機能が、割込み処理の終了時に、割込み信号ラインがノンアクティブであることを確認するように動作する。

【0024】

第11の形態では、第10の形態の中央ブロックのアプリケーション処理機能が、割込み処理の終了時に、割込み信号ラインがアクティブとなっている時は、割込み信号ラインをノンアクティブに戻すように動作する。

【0025】

第12の形態は、第1から第11の形態のブロックの演算機能にデータの一時取り込み機能が設けられており、中央ブロックと、周辺ブロックの疑似マイクロコンピュータ周辺装置との間で大量のデータが送受信される場合に、この大量のデータは中央ブロックとこの演算機能との間でバースト転送により送受信され、演算機能と疑似マイクロコンピュータ周辺装置との間には非バースト転送により少量ずつデータの送受信が行われることを特徴とするものである。

【0026】

第13の形態は、第9の形態の中央ブロックのアプリケーション処理機能により、割込み処理の受け付け終了後にP C Iバスを通じて疑似マイクロコンピュ

タ周辺装置から割込みフラグが取得され、アプリケーション処理機能によって割込みフラグが取得されると、疑似マイクロコンピュータ周辺装置にある当該割込みフラグは、アプリケーション処理機能によりクリアされるものである。

【 0 0 2 7 】

第 1 4 の形態は、第 1 3 の形態の中央ブロックのアプリケーション処理機能により割込みフラグが取得されると、アプリケーション処理機能により、取得した割込みフラグに付随する処理が実行されることを特徴とするものである。

【 0 0 2 8 】

第 1 5 の形態は、第 1 4 の形態の中央ブロックのアプリケーション処理機能により、割込み処理の受け付け終了後に P C I バスを通じて疑似マイクロコンピュータ周辺装置から複数の割込みフラグがアプリケーション処理機能に取得された場合に、アプリケーション処理機能によって優先度の高い割込みフラグが 1 つ選択されてそのフラグに対する処理が実行され、処理終了後に、疑似マイクロコンピュータ周辺装置にある当該処理完了割込みフラグは、アプリケーション処理機能によりクリアされることを特徴としている。

【 0 0 2 9 】

第 1 6 の形態は、第 1 5 の形態のアプリケーション処理機能によって優先度の高い割込みフラグが 1 つ選択されてそのフラグに対する処理が実行された後は、アプリケーション処理機能により、再度 P C I バスを通じて疑似マイクロコンピュータ周辺装置から複数の割込みフラグがアプリケーション処理機能に取得されることを特徴とするのである。

【 0 0 3 0 】

第 1 7 の形態は、第 1 3 から第 1 6 の形態の割込みフラグが、疑似マイクロコンピュータ周辺装置の 1 つのレジスタの連続するアドレスに、まとめて記憶されていることを特徴とするものである。

【 0 0 3 1 】

第 1 8 の形態は、第 1 7 の形態の周辺ブロックが複数個設けられており、各周辺ブロックに設けられた各レジスタには、各周辺ブロックにおいて割込み処理を発生させる各リソースの割込みフラグがそれぞれ格納されており、第 1 番目の周

辺ブロックに設けられたレジスタには、第 1 番目の周辺ブロックにおける各リソースの割込みフラグがそれぞれ格納されていると共に、残りの周辺ブロックにおける各リソースの割込みフラグの有無を示す拡張割込みフラグが、各周辺ブロック毎に格納されていることを特徴とするものである。

【0 0 3 2】

第 1 9 の形態は、第 1 8 の形態の拡張割込みフラグにより、残りの何れかの周辺ブロックのレジスタに割込みフラグが格納されていることが判明した場合には、アプリケーション処理機能により、残りの周辺ブロックの当該レジスタから割込みフラグが取得されることを特徴としている。

【0 0 3 3】

第 2 0 の形態は、第 1 から第 1 9 の形態の周辺ブロックが複数個設けられており、第 1 番目の周辺ブロックのみにフリーランタイムが存在し、第 1 番目の周辺ブロックには、少なくともフリーランタイムのタイマ値に同期して動作するリソースが配置され、残りの周辺ブロックには、フリーランタイムに依存しないリソースのみが配置されていることを特徴としている。

【0 0 3 4】

第 2 1 の形態は、第 2 0 の形態のフリーランタイムのタイマ値に同期して動作するリソースがコンペア、キャプチャであり、フリーランタイムに依存しないリソースが P W M、通信、A D、及びポートであることを特徴としている。

【0 0 3 5】

なお、以上の全ての形態において、中央ブロック、周辺ブロック、インタフェース回路ブロックをそれぞれ汎用のボードから構成することができる。また、本発明のマイクロコンピュータのロジック開発装置は、内燃機関の制御用のマイクロコンピュータに有効に適用できる。

【0 0 3 6】

以上のように構成された本発明のマイクロコンピュータのロジック開発装置によれば、マイクロコンピュータのロジックの開発に伴う課題が解消され、C P U 機能に対しては新規なロジックや次期ロジックを処理するために必要な処理性能を確保することができ、マイクロコンピュータ周辺リソースに対しては、新規シ

システムや次期システムに合わせたリソースを確保することができて、新規ロジックや次期ロジックを実現することが可能な組込み用マイクロコンピュータを短時間で開発することが可能となる。また、本発明のマイクロコンピュータのロジック開発装置は、ロジックの開発に際して繰り返し利用することが可能であるので、開発コストを低減することができる。

【0037】

【発明の実施の形態】

以下添付図面を用いて本発明の実施形態を具体的な実施例に基づいて詳細に説明するが、以下の実施例では、本発明を適用する電子制御機器として、電子制御式内燃機関（内燃機関は以下エンジンと記す）を説明する。

【0038】

図1は従来の電子制御式エンジンの制御システムにおけるECU（電子制御ユニット）1の構成を示すシステム構成図である。電子制御式エンジンでは、エンジン回転数信号や車速信号等のパルス入力、水温センサや吸気温センサ等からのアナログ入力、及びスタータスイッチ、電気負荷スイッチ、シフト位置スイッチやエアコン信号等のデジタル入力がECU1に入力される。ECU1は、これらの入力信号を処理する組込み用マイクロコンピュータ2と、組込み用マイクロコンピュータ2で処理された信号を増幅して出力するECU入出力回路であるドライバ16とを備えて構成される。このECU1から出力されるのは、シフト制御ソレノイドやVVT（可変バルブタイミング）ソレノイドへのアナログ出力、点火信号や燃料の噴射信号等のパルス出力、ISC（アイドル速度制御）用のパルス出力、及び、チェックエンジンランプ、メインリレーやエアコンカット信号等のデジタル信号等である。

【0039】

組込み用マイクロコンピュータ2は、演算処理を行うメモリ9とCPU10、及び入出力（I/O）制御を行う周辺リソースが、1つのパッケージに収納されたものである。周辺リソースには、入力系のリソースと出力系のリソースとがある。図1には、入力系のリソースとして、デジタル信号を扱う入力ポート3とラッチポート4、アナログ入力を扱うA/Dコンバータ5、及びパルス入力を扱う

キャプチャ 6 が示してあり、出力系のリソースとしては、デジタル出力を出力する出力ポート 12、パルス出力を出力する PWM（パルス幅変調器）13 とコンペア 14、及びアナログ出力を入出力するシリアル 15 が示してある。これらの周辺リソースは内部バス 11 によってメモリ 9 及び CPU 10 に相互に接続されている。また、組込み用マイクロコンピュータ 2 の内部には、これらの周辺リソースに加えて、内部タイマ 7 や割り込みコントローラ 8 が設けられている。組込み用マイクロコンピュータ 2 の全ての端子はポートと呼ばれる。

【0040】

電子制御式エンジンの制御システムでは、車両の運転状態を表す各センサやスイッチ類からの信号が ECU 1 に取り込まれる。ECU 1 の入力回路では入力信号が信号処理され、組込み用マイクロコンピュータ 2 に入力される。入力された信号は前述の入力系の周辺リソースで CPU 値に変換され、演算部であるメモリ 9 と CPU 10 では入力信号から車両状態が検出され、車両状態に応じた出力要求信号が作られる。この出力要求信号は前述の出力系の周辺リソースで出力信号に変換され、組込み用マイクロコンピュータ 2 から出力される。ECU 1 の出力回路であるドライバ 16 はこの出力信号に従って車両に装備された各アクチュエータを駆動し、この出力制御の結果が破線で示すように、車両からの入力信号に反映される。

【0041】

図 2 は、図 1 で説明した ECU 1 の自動車（車両）18 への搭載位置を示すものである。ECU 1 は車両 18 のエンジン 19 がマウントされるエンジンルームに搭載される。また、図 2 には、本発明のマイクロコンピュータのロジック開発装置 20 が示してある。本発明のマイクロコンピュータのロジック開発装置 20 は、この図に示すように、車両 18 に搭載された ECU 1 に接続するコネクタを外し、このコネクタに接続コード 20A によって直接接続して使用することができる。

【0042】

本発明のマイクロコンピュータのロジック開発装置 20 は、上位コンピュータ（ホストコンピュータ）27 に接続されている。ホストコンピュータ 20 には表

示器 2 1 と、データを入力するためのキーボード 2 2 が備えられている。S 1, S 2 はそれぞれロジック装置 2 0 とホストコンピュータ 2 7 を動作させるための電源スイッチである。表示器 2 1 は、マイクロコンピュータのロジック開発装置 2 0 の状態をモニタすることができ、キーボード 2 2 からは、マイクロコンピュータのロジック開発装置 2 0 の設定を変更するための入力を行うことができる。

【0 0 4 3】

なお、本発明のマイクロコンピュータのロジック開発装置 2 0 は、このように車両 1 8 に直接接続して使用することができる他に、パーソナルコンピュータ 2 4 の制御によって動作して、車両の色々な運転状況を擬似的に発生することができる車両の運転状況発生装置 2 3 に接続すれば、車両 1 8 が無い状態でも、電子制御式エンジン用の組み込みマイクロコンピュータのロジックを開発することができる。

【0 0 4 4】

図 3 は、図 2 のように接続して使用することが可能な本発明のマイクロコンピュータのロジック開発装置 2 0 のシステム構成を、従来の電子制御機器（ここでは電子制御式エンジン）用の E C U 1 の構成と比較して示すものである。前述のように、E C U 1 には、組込み用マイクロコンピュータ（図には組込み用マイコンと略記）2 と、ドライバ 1 6 から構成される E C U 入出力回路 2 8 があり、E C U コネクタ 2 9 で車両側の電子機器と接続されている。また、組込み用マイクロコンピュータ 2 の中には、メモリ 9 に格納されていて C P U 1 0 によって読み出されて使用されるソフトウェア（エンジン制御アプリケーション：図には E N G 制御アプリと略記）2 5 と、マイクロコンピュータ周辺リソース（図にはマイコン周辺と記載）2 6 とがあり、内部バス 1 1 で相互にデータの送受信ができるようになっている。

【0 0 4 5】

一方、図 2 で説明したように、この E C U 1 に置き換えて使用する本発明のマイクロコンピュータのロジック開発装置 2 0 は、この実施例では、中央ブロックであるマザーボード 3 0、周辺ブロックであるコアボード 4 0、及びインタフェース回路ブロックである I F ボード 5 0 の 3 つのボードから構成されている。マ

ザーボード 3 0 とコアボード 4 0 が E C U 1 の組込み用マイクロコンピュータ 2 に対応するものであり、I F ボード 5 0 が E C U 1 の E C U 入出力回路 2 8 に対応するものである。そして、マザーボード 3 0 とコアボード 4 0 とは、高速のバスインタフェースである P C I バス 3 9 で接続されている。

【 0 0 4 6 】

マザーボード 3 0 には、後述するメモリに格納されていて C P U によって読み出されて使用されるソフトウェア（E N G 制御アプリケーション） 3 1 と、P C I バス 3 9 を通じて通信を行うための P C I 通信ソフトウェア（図には通信ソフトと略記） 3 2 が設けられている。P C I バス 3 9 を用いた P C I 通信処理は、疑似マイクロコンピュータ周辺リソース（図には疑似マイコン周辺と略記） 4 2 と送受信するデータを、P C I バス 3 9 に載せる通信処理である。このマザーボード 3 0 には、次期の E C U を開発するに当たっては、次期 E C U の先行ロジックの開発に耐え得る演算性能、メモリ容量を備えさせることが重要である。

【 0 0 4 7 】

なお、現状のエンジン制御用のマイクロコンピュータの性能は、C P U が 6 4 M H z 、メモリが 1 M バイト程度であるので、パソコン等に用いられている汎用のものを用いれば、十分すぎる性能であるといえ、長期間にわたって何度でも使用することが可能となる。

【 0 0 4 8 】

また、コアボード 4 0 は C P U とメモリを含み、前述の P C I バス 3 9 と通信を行うための P C I 通信ソフトウェア 4 1 と、組込み用マイクロコンピュータ 2 のマイクロコンピュータ周辺リソース 2 6 に対応する疑似マイクロコンピュータ周辺（F P G A : Field Programmable Gate Array） 4 2 とがあり、内部バス 4 3 で相互にデータの送受信ができるようになっている。ここではポートは F P G A 4 2 の全ての端子である。

【 0 0 4 9 】

I F ボード 5 0 には、E C U 1 の E C U 入出力回路 2 8 に対応する E C U 入出力回路 5 1 と、E C U コネクタ 2 9 とが設けられている。E C U 入出力回路 5 1 は、標準回路ブロック単位で独立させ、その組み合わせで構成し、入出力回路の

変更に対して柔軟に対応できるようにする。

【0050】

図4は、図3のマザーボード30とコアボード40の、ハードウェアの構成の一実施例を示すものである。マザーボード30には、図3で説明したソフトウェア（ENG制御アプリケーション）を記憶するための記憶容量が大きいメモリ31、汎用の高性能のCPU（例えば、動作周波数が850MHz）33、内部タイマ35、PCIバス39に接続するPCIバスインタフェース36、及びこれらを相互に接続する内部バス37がある。

【0051】

コアボード40には、PCIバス39に接続するPCIバスインタフェース44、マザーボード30に搭載されたCPU33よりも低グレードのCPU45、マイクロコンピュータ周辺の機能と同等の機能を実現する疑似マイクロコンピュータ周辺（FPGA）42、内部バス43、PCIバスに接続する共有メモリ46、及び内部バス43に接続する内部メモリ47がある。また、コアボード40に搭載するCPU45は、PCI通信処理を行うことができる程度の処理能力（例えば、汎用32ビットCPUで動作周波数が16MHz程度）であれば良い。コアボード40の機能は、従来のECU1におけるエンジン制御アプリケーション（ソフトウェア）25とマイクロコンピュータ周辺26とが送受信するデータをPCIバス39を経由して受け、FPGA42へ受け渡すことである。

【0052】

IFボード50に接続される各FPGA42はソフトウェアで組むことができ、マイクロコンピュータ周辺の変更に対応させることができる。即ち、チャンネル数を増やしたい場合や、新しい機能のリソースを追加したい等の場合に対応させることができる。

【0053】

図5は、本発明のマイクロコンピュータのロジック開発装置20における、ECU1のソフトウェアの構成（一部ハードウェアも記載）を示すものである。ECU1には、マイクロコンピュータコア相当の機能を実現するマザーボード30、マイクロコンピュータリソース相当の機能を実現するコアボード40、及びE

C U 1 のハードウェア相当の機能を実現する I F ボード 5 0 がある。E C U 1 のハードウェアとは、マイクロコンピュータ以外の電気回路のことである。

【0054】

マザーボード 3 0 及びコアボード 4 0 には、マイクロコンピュータリソースに接続するバス相当の機能を実現する I / O ドライバ (マザーボード側 I / O ドライバ 3 0 D とコアボード側 I / O ドライバ 4 0 D) が実装されており、その間は P C I バス 3 9 で接続されている。P C I バス 3 9 には 1 系統の割込み信号ライン A があり、この信号ライン A は、コアボード 4 0 からマザーボード 3 0 への割込み要求を発行することができる。

【0055】

また、P C I バス 3 9 は、マザーボード 3 0、コアボード 4 0 に搭載されたマイクロコンピュータ 4 0 M にあるコア側 I / O ドライバ 4 0 D、及び F P G A 4 2 に相互に接続されている。そして、本発明における P C I バス 3 9 は、経路 B で示すマザーボード 3 0 と F P G A 4 2 の間の直接アクセスと、経路 C で示すマザーボードと F P G A 4 2 の間のコアボード側 I / O ドライバ 4 0 D を介したアクセスの、2 通りの方法でマザーボード 3 0 とコアボード 4 0 の間のデータの送受信を行うことができる。

【0056】

このように、本発明では、図 5 に示すように、P C I バス 3 9 によって、あたかもマザー側 I / O ドライバ 3 0 D と F P G A 4 2 との直接アクセスが実現できるが、実際にはハードウェアとしてコアボード 4 0 の中にバスコントローラがある。これを図 6 を用いて説明する。図 6 に示すように、コアボード 4 0 にあるマイクロコンピュータ 4 0 M には、内部メモリ 5 8、プログラム 5 9、及びバスコントローラ 6 0 がある。バスコントローラ 6 0 は P C I バス 3 9 によってマザーボード 3 0 に接続され、コアボード 4 0 の内部バス 5 6 によって F P G A 4 2 にあるレジスタ 6 1 に接続されている。また、バスコントローラ 6 0 は、マイクロコンピュータ 4 0 M の内部バス 5 9 によって内部メモリ 5 8 に接続されている。

【0057】

ハードウェア的にはバスは一重構造である。よって、バスコントローラ 6 0 に

よって、P C Iバス39と内部バス56を経由してマザーボード30からF P G A 42のレジスタ61にデータが直接送られると、マザーボード30とレジスタ61とはハードウェア的につながっているのと同じになる。即ち、内部的な操作としては、バスコントローラ60を介してデータが送られるが、外部的な操作としては、F P G A 42にマザーボード30から直接データが書き込まれたり、F P G A 42から直接データがマザーボード30に読み出されたりするのと同じである。このため、あたかも図5に示される1つのP C Iバス39でマザーボード30とF P G A 42の間にデータの送受信が行われているようになる。

【0058】

バスコントローラ60はデータの流れの切り換えを行うことができ、バスコントローラ60に入力されたデータは、一旦内部バス57を介して内部メモリ58に書き込むことができる。よって、経路Bは、P C Iバス39、バスコントローラ60、内部バス56を介して、マザーボード30とF P G A 42との間でデータを送受信するものであり、経路Cは、P C Iバス39、バスコントローラ60、内部バス56の経路の途中でプログラム59が働いて、一旦データが内部メモリ58に書き込まれるものである。このプログラム59の部分が図5のコア側I/Oドライバ40Dに相当する。

【0059】

図5に戻って、E C U 1のハードウェアに相当するI Fボード50には、ポート割付変換ボード52と複数個の標準回路53、及び機能ボード55があり、これらの組み合わせにより任意の数の入出力回路を実現できる。また、54は図3に示したE C Uコネクタ29に相当するものである。I Fボード50とコアボード40とは、マイクロコンピュータポートに相当するハーネス49で接続されている。

【0060】

機能ボード55は、エンジン制御アプリケーション31から切り出された機能を実現するために設けられており、標準回路53との相違は、マイクロコンピュータが内蔵されている点である。通常の標準回路53はオンオフの機能だけであるが、機能ボード55にはマイクロコンピュータがあるので通信機能がある。即

ち、機能ボード55は、FPGA42との間でDMA通信によってデータを送受信することができ、エンジン制御アプリケーション31から出力された機能を実現することができる。このような構成を利用したデータの送受信が本発明の特徴である。

【0061】

なお、電子制御機器が電子制御式エンジンの場合は、マザーボード30には、エンジン制御アプリケーション31と、このアプリケーション31とFPGA42の間のI/O情報の伝達を行うI/Oドライバ30Dの2つの大きなソフトウェアブロックがある。エンジン制御アプリケーション31には更に、図示はしないが、演算のみを行うブロックと、I/O操作と演算処理が混在するブロックとがある。I/O操作と演算処理が混在するブロックは、実ECUに実装されているソフトウェアが略これに相当する。演算のみを行うブロックは、追加検討ロジックであり、I/O操作と完全に切り分けられている。I/Oドライバは、前述のように、マザーボード側I/Oドライバ30Dとコアボード側I/Oドライバ40Dに分かれており、PCIバス39経由でボード間の同期をとり、I/O情報の伝達を行う。

【0062】

エンジン制御アプリケーション31の演算のみを行うブロックは時間同期処理（時間系割り込み処理）のみを行うが、I/O操作と演算処理が混在するブロックは時間同期と非時間同期処理（時間系割り込み処理）の2つの処理を行う。時間同期処理は外部状態に依存せず、一定の時間間隔で定期的に行う処理である。一方、非時間同期処理は、外部状態、即ち、エンジンの運転状態に依存して発生する処理である。この非時間同期処理には、例えば、エンジンの回転信号、車速信号、燃料の噴射タイミング信号、点火タイミング信号等による割り込み処理がある。

【0063】

ここで、このような外部状態によりランダムに変化する事象をイベントと呼ぶこととすると、非時間同期処理は、外部状態によりイベント情報を検出して、そのイベントに同期して行う処理ということができる。

【0064】

マザーボード30のエンジン制御アプリケーション31において、前述のような時間同期処理と非時間同期処理とを行うために、図4で説明したマザーボード30のCPU33は仮想マイクロコンピュータ周辺として機能する。この仮想マイクロコンピュータ周辺には、時間系割り込みと非時間系割り込みを発生させる仮想割り込みコントローラと、仮想I/Oレジスタが設けられている。また、PCI通信ソフトウェアとしてのマザーボード側のI/Oドライバ30Dは、データとして、割り込みイベント情報と、I/Oレジスタデータとを扱う。

【0065】

一方、コアボード40側のFPGA（疑似マイクロコンピュータ周辺）42には、図3に示した従来のマイクロコンピュータ周辺26と同様に、ポート、ラッチ、PWM、シリアル、コンペア、及びキャプチャの各機能が設けられている。また、PCI通信ソフトウェアとしてのコアボード側I/Oドライバ40Dは、データとして、割り込みイベント情報と、I/Oレジスタデータとを扱う。更に、コアボード40には、イベントの流れを発生させるタイマが内蔵されている。

【0066】

ここで、制御対象が車両のエンジンである場合に、前述のイベント情報とデータとが流れる本発明のマイクロコンピュータのロジック開発装置20が実施する、マザーボード30とコアボード40との間のデータの送受信について、図7に示すタイミングチャートと、図8に示すソフトウェアの構成図を用いて全体的な流れを説明し、その後、図7に示すタイミングチャートにおける各処理の詳細について、具体的な実施例に基づいて個々に説明する。

【0067】

なお、以下の実施例では、主として、イベントの流れが実線で示され、データの流れが点線で示されている。

(1) マザーボード30とコアボード40との間の通信方法

図7はマザーボード30とコアボード40との間の割り込み要求の発生に伴うデータの流れを示すタイミングチャートである。図7には、マザーボード30、PCIバス39、及びコアボード40における処理が、時間の経過と共に示してあ

る。なお、コアボード 4 0 は図示のように複数個使用することができるが、コアボード 4 0 を複数個使用する場合の制御については後述し、この実施例では 1 つのコアボード 4 0 を使用した場合について説明する。また、ここでは、コアボード 4 0 における割込み発生待ちの開始 S から、アプリケーション処理完了通知の受信 E まだが 1 つの処理サイクルであり、以後はこの処理サイクルが繰り返されるので、1 つの処理サイクルについてのみ説明を行う。

(1-1) 割込みタイミング

マザーボード 3 0 のエンジン制御ソフトウェア（アプリケーション処理）は、外部イベント（FPGA 4 2 内のタイマ、エンジン回転信号、車速信号、通信信号等）が起点となって動作する。外部イベントの発生は、ロジック開発装置では、割込みフラグの発生に置き換えることができる。そこで、エンジン制御ソフトウェアの中に外部割込フラグによって起動する割込みルーチンを設けておき、外部割込フラグの入力があると、この割込みルーチンが起動するようになっている。この割込みルーチンは、信号に対して割り込まれてハードウェア的に起動されるので、割込みハンドラと呼ばれる。

【0068】

マザーボード 3 0 には割込み機構がなく、イベントの伝達方法としては P C I 割込み 1 系統（図注の A で示す信号ライン）しかないため、割込みタイミングはコアボード 4 0 側のマイクロコンピュータで各割込みフラグを監視し、何らかの割込みフラグがアクティブになった場合に、この 1 系統の割込み信号ライン A を使用して信号を送り、マザーボード 3 0 に割込みタイミングとして通知するのである。

【0069】

割込み要因は多数あるが、割込み要因の有無を監視して取り込むのが F P G A 4 2 である。割込み要因が 1 つでも存在すれば、F P G A 4 2 はその値を読み、コアボード 4 0 側の I / O ドライバ 4 0 D が動作し、割込みタイミングが信号ライン A を通じてマザーボード 3 0 に通知される。すると、F P G A の割込みフラグが I / O ドライバ 3 0 D に直接読み込まれ、マザーボード 3 0 のアプリケーション処理が開始される。そして、どの割込みフラグが立っているかに応じて、そ

のフラグに対するアプリケーション処理が順次行われる。割込みタイミングが信号ラインAを通じてマザーボード30に通知された後は、コアボード40側のI/Oドライバ40DはPCI割込み禁止状態となる。

(1-2) I/Oデータ

I/Oデータは、基本的にマザーボード30がPCIバス39を経由してFPGA42のレジスタを直接読むことによって取得され、逆に、マザーボード30がPCIバス39を経由してFPGA42のレジスタに直接書き込むことによって設定される。大量のデータを送受信する場合は、バースト転送が行われ、通信効率が上げられる。ポート、ラッチの速度が要求されないビットデータは、I/Oドライバの前処理、後処理で一括して送受信される。

【0070】

アプリケーション処理では、コアボード40のI/Oドライバ40Dにアクセスして或るポートの中を見る処理、出力をセットする処理等が、アプリケーション処理を実行しながら行われ、図7にいくつかの破線で示すように、直接FPGA42からデータを読み込んだり、直接FPGA42にデータを書き込んだりすることが行われる。

【0071】

アプリケーション処理が終了すると、図5に破線の経路Bで示す最後の処理が行われ、割込み処理の終了がコアボード40側のFPGA42のレジスタに書き込まれる。すると、コアボード40内に割込み処理が発生し、アプリケーション処理完了通知の受信Eが太線で示すようにI/Oドライバ40Dに伝えられる。この結果、コアボード40側のI/Oドライバ40DはPCI割込み発生許可状態に移行し、割込みの発生待ちの開始Sとなり、割込みフラグの常時監視状態となる。ここで、何らかの割込み要因が存在していたら、コアボード40側のI/Oドライバ40Dが動作し、割込みタイミングがラインAを通じてマザーボード30に通知され、次の割込み処理が行われる。以後はこの繰り返しである。

【0072】

以上のような本発明におけるマザーボード30側のアプリケーション処理では、マザーボード30が直接FPGA42に対してアクセスしてデータの送受信を

行うので、コアボード40において入力情報を集約して取得後に、出力要求を集約してまとめてマザーボード30に送る方法に比べて、I/Oドライバに対する指示の遅れが小さく、応答性が良い。

【0073】

図8は図7のマザーボード30における入力情報の取得、アプリケーション処理、及び出力要求確定の処理の詳細を示すものである。図7に示したPCIバス39を通じてPCI割込みがマザーボード側のI/Oドライバ30Dに入力されると、I/Oドライバ30Dは先ずFPGA42から割込みフラグを取得してこれを割込みコントローラに入力する。割込みフラグを取得した後は、この割込みフラグのクリアを行ってから入力情報を一括取得する。これは、入力情報を取得してから割込みフラグを取得すると、割込みフラグの取得によって変化する入力情報があるかもしれないからである。よって、本発明では割込みが確定した後にその時の最新情報を読むことで、確実に必要なタイミングの値を読む。即ち、割込みコントローラで割込み要因の確定をしておいてから、割込みハンドラの処理に移る直前で入力情報をI/Oドライバ前処理として一括取得するのである。一括取得する入力情報は、ポートレベル、キャプチャ値、受信データ、AD変換値等の入力データを含む。

【0074】

一括取得するデータは、取得スピードが不要で、1ビットずつに分かれているものであり、このようなデータは、効率良く取得するために一括して取得するのである。データを一括取得した後は、仮想I/Oレジスタ(RAM)38に入力情報を設定しておく。これは、仮想I/Oレジスタ38を介してマザーボード30が、あたかも本当のマイクロコンピュータの端子に接続されているような形でデータを送受信するためである。

【0075】

このようにして、割込みフラグの取得と入力情報の一括取得が行われた後、割込みコントローラは割込みフラグを見てそれに付随するアプリケーション処理を起動する。アプリケーション処理は割込みハンドラから開始される。

【0076】

アプリケーション処理にはアプリケーション層 L1、2つのハードウェア依存層 L2、L3、及びマイクロコンピュータ依存層 L4 があり、これらの層で必要な演算が実行される。この演算の中で I/O 処理があり、入力情報を取るという命令がきた場合は、本発明では直接 FPG A 42 からデータをとる。マイクロコンピュータ依存層 L4 と FPG A 42 との間に描かれた破線、及びマイクロコンピュータ依存層 L4 と仮想レジスタ 38 との間に描かれた破線が、図 7 で説明したアプリケーション処理から出たり入ったりする破線の処理に相当する。取得したデータは仮想 I/O レジスタ 38 に一旦蓄積しておく。出力設定があった場合は出力設定を仮想 I/O レジスタ 38 に書き込むと同時に、実際に命令を出すように FPG A 42 に出力する。

【0077】

この場合、仮想 I/O レジスタ 38 に設定済（ポート入力情報は一括して先頭で取得済である）のものは、仮想 I/O レジスタ 38 からデータを読み取るだけである。即ち、色々な場合があり、データを FPG A 42 から直接読み取る場合と、最初に一括して設定したデータを読み取るだけのものなどがある。また、図 8 に記載したマイクロコンピュータ依存層 L4 と仮想 I/O レジスタ 38 あるいは FPG A 42 とのデータの送受信は一例であり、このようなデータの送受信は何回行っても良い。また、マイクロコンピュータ依存層 L4 と FPG A 42 とのデータの直接アクセスはできるが、あまり頻繁にアクセスを行うと PCI バス 39 の負荷が大きくなるので、一括してできる送受信は一括して行い、それ以外の送受信は、直接アクセスして応答性を向上させるという混在処理が行われる。一例として、入力ポート、出力ポート、ラッチポートは一括して処理を行い、それ以外は直接データを FPG A 42 と送受信している。

【0078】

このようにして、アプリケーション処理が終了すると、I/O ドライバ 30 D の後処理として、出力要求の一括設定と、コアボードへの PCI 割込み許可要求が実行される。出力要求の一括設定では、仮想 I/O レジスタ 38 と FPG A 42 に出力設定が行われ、コアボード 40 への PCI 割込み許可要求では、FPG A 42 に対して割込み許可要求が行われる。I/O ドライバ 30 D の後処理では

、得られたポートレベル、コンペア、PWM、送信データ、通信起動データ、A/D起動データ等の出力情報がP C Iバス39を経由してコアボード40に出力される。この後、I/Oドライバ30Dは次回のP C I割込みの発生待ちとなる。

(2) マザーボード30とコアボード40との間の各処理の詳細

以上、マザーボード30とコアボード40との間の割込み要求の発生に伴うデータの流れを全体的に説明したが、ここで、図7に示したマザーボード30、P C Iバス39、及びコアボード40における処理の個々の処理について、更に詳しく説明する。ここで説明するのは、図7に丸付き数字1から5で示す以下の処理の詳細である。(2-1)丸付き数字1で示すパワーオンシーケンス、(2-2)丸付き数字2で示すP C I割込み信号の処理、(2-3)丸付き数字3で示すP C Iアクセス方法、(2-4)丸付き数字4で示す割込みコントローラの動作、及び(2-5)丸付き数字5で示す複数のコアボードの同期処理。

(2-1) パワーオンシーケンス

パワーオンシーケンスとは、図2に示した本発明のエンジン制御システムにおいて、ロジック開発装置20の電源スイッチS1とホストコンピュータ27の電源スイッチS2とがオンされた後に、イグニッションスイッチI G S Wがオンされた場合のシステムへの電源の供給シーケンスである。パワーオンシーケンスを説明するために、図9には、ホストコンピュータ27、オペレーティングシステムと初期化ルーチンとメインルーチン及びP C I割込みルーチンを備えたマザーボード30、P C I割込み発行機能を備えたコアボード40、電源I C 65とA N D回路66及びマイクロコンピュータ67を備えた機能ボード55、バッテリー62、イグニッションスイッチI G S W、及び比較器63、64が図示されている。

【0079】

初期化ルーチンは、ホストコンピュータ27からデータが転送される(ダウンロードされる)毎に1回起動される。また、メインルーチンは、システムの起動中に1 m s 毎に定期的に起動される。P C I割込みルーチンは、システムの起動中の割込み要求発生毎に起動される。この割込み要求は、E C U 1のソフトウェアのパワーオン以外の全てのソフトウェアの起動要求である。このルーチンによ

って、ECU 1 の全てのメイン処理が動作する。

【0080】

パワーオンシーケンスにおける電源は、実際のエンジン制御システムと同様にバッテリー 6 2 である。このバッテリー 6 2 の出力電圧 + B (= 1 2 V) は 2 つに分岐され、一方はイグニッションスイッチ I G S W を介して比較器 6 3 の第 1 の入力端子に接続され、他方はそのままもう 1 つの比較器 6 4 の第 1 の入力端子に接続されている。比較器 6 3 の第 1 の入力端子への電圧は、機能ボード 5 5 の電源 I C 6 5 へも供給される。これら 2 つの比較器 6 3 , 6 4 の第 2 の入力端子には、バッテリー 6 2 の出力電圧 + B よりも低い + 5 V の固定電圧が入力されている。比較器 6 3 , 6 4 の出力端子は、マザーボード 3 0 の直接入力信号を受けることができるプリンタポート 6 8 に接続されている。

【0081】

比較器 6 3 は、イグニッションスイッチ I G S W のオンにより出力端子がハイレベルになり、比較器 6 4 の出力端子はバッテリー 6 2 が外されない限り常にハイレベルとなっている。よって、比較器 6 3 , 6 4 の出力が入力されるメインルーチンでは、比較器 6 3 の出力がハイレベルになったことによりイグニッションスイッチ I G S W のオンを検出することができると共に、比較器 6 4 の出力がローレベルになったことによりバッテリー 6 2 の取り外しを検出することができる。

【0082】

マザーボード 3 0 上のオペレーティングシステムは、エンジン制御において、イグニッションスイッチ I G S W のオンにより動作し、イグニッションスイッチ I G S W のオフによって動作を停止する。実際のマイクロコンピュータでは、信号を受けてマイクロコンピュータのリセットが解除されて動き出すというシーケンスをとるが、ここではイグニッションスイッチ I G S W のオンオフに連動させている。

【0083】

機能ボード 5 5 にはマイクロコンピュータ 6 7 が搭載されており、イグニッションスイッチ I G S W がオンされても、マイクロコンピュータ 6 7 はこれに同期してオンしないようになっている。機能ボード 5 5 の A N D 回路 6 6 の第 1 の入

力端子には、マザーボード 3 0 からの第 1 のイニシャル信号 I N T 1 が入力され、第 2 の入力端子には電源 I C 6 5 からの第 2 のイニシャル信号 I N T 2 が入力される。電源 I C 6 5 にはイグニッションスイッチ I G S W がオンの状態で、バッテリー 6 2 からの電圧 + B が入力される。第 1 のイニシャル信号 I N T 1 は、マザーボード 3 0 と処理開始タイミングの同期を取るためのものであり、第 2 のイニシャル信号はイグニッションスイッチ I G S W のオンの後の、後述するパワーオンリセット時間を確保するためのものである。

【 0 0 8 4 】

なお、実際のマイクロコンピュータ 6 7 にはバッテリー 6 2 の電源が常に供給されているバックアップ R A M (B - R A M) 7 0 があってイグニッションスイッチ I G S W がオフされてもデータを記憶している。本発明の構成には実際にはバックアップ R A M 7 0 は設けられないこともあるが、その機能は実現している。また、バッテリーが外されてもデータを覚えている E E P R O M と同等のことは実現する方法がパワーオンシーケンスには盛り込まれている。

【 0 0 8 5 】

ここで、このパワーオンシーケンスの動作を、図 1 0 に示す処理手順を用いて説明する。図 1 0 には、図 9 のように構成されたマザーボード 3 0 、コアボード 4 0 、及び機能ボード 5 5 の各処理におけるステップが記載されている。

【 0 0 8 6 】

ステップ 1 は、図 2 のように構成されたシステムにおいて作業者がロジック開発装置 2 0 とホストコンピュータ 2 7 の電源をオンした状態である。この作業者のシステム電源のオンにより、マザーボードとコアボードの電源がオンするが、機能ボードの電源はオフのままである。この状態は、ロジック開発装置 2 0 とホストコンピュータ 2 7 の本体の電源投入のことであり、イグニッションスイッチ I G S W のオンではない。

【 0 0 8 7 】

ステップ 2 では、処理 1 の電源のオンにより、1 m s 毎に電源のオンを検出している図 9 のメインルーチンが起動し、ホストコンピュータ 2 7 からアプリケーションのソフトウェアがマザーボード 3 0 にダウンロードされ、ロジック開発装

置 2 0 のマザーボード 3 0 が起動する。このとき、コアボード 4 0 も起動するが、コアボード 4 0 の P C I 割込みはロック状態である。

【 0 0 8 8 】

ステップ 3 では、図 9 のマザーボード 3 0 の初期化ルーチンにより、ポートの初期値が設定される。この処理設定は 1 回きりであり、その内容は、(1) コンペアをイミディエート出力によって設定する、(2) P W M を 0 % 出力に設定する、(3) 第 1 のイニシャル信号 I N T 1 をローレベルに設定する、(4) メインルーチンにステート 1 を設定する、である。

【 0 0 8 9 】

ステップ 4 ではステート 1 の最初の処理 (1) が行われる。ステート 1 の処理 (1) では、イグニッションスイッチ I G S W がオンされるのを待つ。

【 0 0 9 0 】

そして、ステップ 5 において作業者が疑似車両のイグニッションスイッチ I G S W をオンすると、ステート 1 の処理 (2) でイグニッションスイッチ I G S W のオンが検出され、機能ボード 5 5 の電源がオンする。この状態は、図 9 におけるイグニッションスイッチ I G S W がオンされた状態で、バッテリー 6 2 から電圧 + B が比較器 6 3 に印加され、比較器 6 3 からのハイレベルの出力で、イグニッションスイッチ I G S W のオンがマザーボード 3 0 によって検出された状態である。

【 0 0 9 1 】

イグニッションスイッチ I G S W がオンされると、機能ボード 5 5 の電源 I C 6 5 に + B が供給される。この後のステート 1 の処理 (3) では、ステート 2 が設定される。なお、電源 I C 6 5 には、電源 + B が供給されてもすぐには第 2 のイニシャル信号 I N T 2 をハイレベルにしない予め定められたパワーオンリセット時間が設定されている。従って、電源 I C 6 5 に電源 + B が供給されても、第 2 のイニシャル信号 I N T 2 はローレベルのままである。

【 0 0 9 2 】

ステップ 6 のステート 2 の処理 (1) では、メモリの復元、即ち、保存してあるファイル (ホストコンピュータ内のメモリ、あるいはマザーボード周辺メモリ

) を復元すること (読み出し) が行われる。すなわち、前回のイグニッションスイッチ I G S W がオフされた時に保存したファイルが読み出される。なお、この時、バッテリー外れがあった場合は、データは初期化される。続くステート 2 の処理 (2) では、E E P R O M のデータ (制御ソフトウェアで演算された学習値データで、車両の走行をくり返す度に最適値に補正されるデータ) が読み出されて復元される。そして、ステート 2 の処理 (3) ではステート 3 が設定される。ステップ 6 では機能ボードの電源 I C はパワーオンリセット時間内であり、第 2 のイニシャル信号 I N T 2 はローレベルのままである。

【 0 0 9 3 】

ステップ 7 のステート 3 の処理では、処理 (1) で機能ボードの電源 I C はパワーオンリセット時間の経過待ちを行い、ステップ 8 のステート 3 の処理 (2) で機能ボードの電源 I C はパワーオンリセット時間の経過を検出する。パワーオンリセット時間が経過すると、機能ボード 5 5 の電源 I C 6 5 から出力される第 2 のイニシャル信号 I N T 2 がハイレベルに変化する。

【 0 0 9 4 】

ステップ 9 のステート 3 の処理 (3) では、マザーボード 3 0 からコアボード 4 0 を通じて機能ボード 5 5 に出力される第 1 のイニシャル信号 I N T 1 がハイレベルに設定される。すると、機能ボード 5 5 の A N D 回路 6 6 の 2 つの入力端子への入力が共にハイレベルになるので、A N D 回路 6 6 の出力がハイレベルになり、機能ボード 5 5 のマイクロコンピュータ 6 7 のリセットが解除されてオンし、プログラムがスタートする。

【 0 0 9 5 】

このように、機能ボード 5 5 はイグニッションスイッチ I G S W のオンの後にすぐに立ち上げるのではなく、ある程度の時間を置いて立ち上げる。即ち、マザーボード 3 0 は、イグニッションスイッチ I G S W のオンからの経過時間がパワーオンリセット時間になった後に、第 1 のイニシャル信号 I N T 1 をハイレベルにして機能ボード 5 5 を動作させるようにし、機能ボード 5 5 とマザーボード 3 0 との同期を取る。この後の処理 (4) では通信以外の機能ボード 5 5 の初期化処理が行われ、処理 (5) でステート 4 が設定される。

【 0 0 9 6 】

ステップ 1 0 のステート 4 の処理では、処理（1）で機能ボード 5 5 の初期化処理が行われ、この機能ボード 5 5 の初期化処理の間、マザーボード 3 0 は待機している。ステート 4 では、機能ボード 5 5 がマイクロコンピュータ 6 7 の動作によって複雑な処理を行うので、機能ボード 5 5 のマイクロコンピュータ 6 7 のリセットを解除して動作し始めると、この中で色々な初期設定が必要となるので、この時間を考慮してマザーボード 3 0 は待機している。

【 0 0 9 7 】

機能ボード 5 5 の複雑な処理（インテリジェントな処理）とは、例えば、出力処理（フィードバック処理）、スロットル弁開度の制御（開度指令値に対して電流フィードバックを行って、指令値になるようにスロットル弁開度を調整する制御）等である。即ち、機能ボードはアプリケーションからの要求値に対してハード的に調整する機能を有している。このための機能ボード 5 5 の初期化がステート 4 の処理（1）で行われる。ステップ 1 1 の処理（2）では、機能ボード 5 5 の内部の処理の完了を、機能ボード 5 5 の初期化時間の経過を見ることによって行う。

【 0 0 9 8 】

機能ボード 5 5 の初期化処理時間が経過すると、処理（3）で機能ボード 5 5 とマザーボード 3 0 との初期化通信が起動され、DMA 通信が開始される。即ち、機能ボード 5 5 は、リセットを解除してプログラムをスタートして初期化処理を開始し、初期化処理が終了すると通常の処理モードに移るので、それを待って DMA 通信が開始される。機能ボード 5 5 はメイン処理を受信する。ここではテスト用の DMA 通信を行う。通常の処理モードのループに入るとデータの頻繁な送受信が行われるが、処理に入る前に正しく通信できるかどうかのチェック用の通信を行う。処理（5）ではステート 5 を設定する。

【 0 0 9 9 】

図 1 1 に移って、ステップ 1 2 のステート 5 の処理では、マザーボード 3 0 と機能ボード 5 5 との初期化通信処理完了待ちを行う。機能ボード 5 5 側では受信処理を行い、受信が終了すると送信データを作成し、次のステップ 1 3 でこれを

送信する。この機能ボード 55 からの送信データの受信により、マザーボード 30 は機能ボード 55 の初期化完了を検知し、これが終わるとステート 6 に移る。なお、機能ボード 55 側の受信処理が完了しないうちにタイムアウトした場合は、ステート 4 に戻る。

【0100】

ステップ 14 ではステート 6 の処理を行う。ステート 6 では、まず、処理 (1) で機能ボード 55 の初期化完了を検知すると、ステップ 15 の処理 (2) でコアボード 40 のロック解除の要求を行う。コアボード 40 のロック解除の要求を行った後、コアボード 40 において割込み要因が発生していたら受け取る。それまでは受け付けない。コアボード 40 としては、最初の電源オンでは P C I 割込みはロック状態なので、ロックが解除されたら割込みフラグを一旦クリアして通信が発生させる。この後の処理 (3) でステート 7 が設定される。

【0101】

以上のステップ 1 からステップ 15 まだが、パワーオンシーケンスの処理である。ステップ 16 からステップ 20 までの処理は、パワーオフシーケンスの処理を示すものである。パワーオフシーケンスにはステート 7 とステート 8 を行うステップ 16 からステップ 20 がある。ステップ 16 からステップ 18 においてはマザーボード 30 とコアボード 40 との間で P C I 割込み処理が繰り返し行われる。そしてこの割込み処理平行して、ステート 7 では処理 (1) でイグニッションスイッチ I G S W のオフの待機状態となる。

【0102】

ステップ 17 のステート 7 の処理 (2) でイグニッションスイッチ I G S W のオフが検出されると、機能ボード 55 の電源がオフされ、処理 (3) でステート 8 が設定される。

【0103】

ステップ 18 のステート 8 の処理 (1) ではコアボード 40 のロックが要求され、コアボード 40 の P C I 割込みロックが行われる。続くステップ 19 のステート 8 の処理 (2) では、マザーボード 30 のメモリ、E E P R O M のデータを保存し、コアボード側で P C I 割込みが禁止される。

【0104】

最後のステップ20におけるステート8の処理(3)ではポートの初期値設定が行われる。この初期設定はステップ3の初期設定と同じであり、処理完了後、ステート1が設定されて前述のステップ3からステップ19が繰り返される。

(2-2) P C I 割込み信号の処理

図12はP C I 割込み信号(図7の信号ラインAの信号)の処理の詳細を示すものである。P C I 割込み信号はイベントが何もない時にはハイレベルの信号となっている。この時、コアボード40のマイクロコンピュータは割込み発生待ち状態を継続している。そして、何かイベントが発生すると、コアボード40のマイクロコンピュータがP C I 割込みアクティブ出力を信号ラインAに出力し、信号ラインAのP C I 割込み信号がローレベル(アクティブ)となる。この後、コアボード40のマイクロコンピュータは、P C I 割込み禁止区間を設定する。また、前述のP C I 割込みラインAのローアクティブ信号は、F P G A 4 2 のP C I 割込み信号レジスタにオン出力として記録される。

【0105】

ローレベルのP C I 割込み信号は信号ラインAによってマザーボード30に伝えられ、マザーボード30ではこのローレベルのP C I 割込み信号を受けて、P C I 割込みの受付が行われる。マザーボード30では、P C I の割込みを受け付けたら、P C I 割込み処理が行われ、最初の処理でP C I 割込みノンアクティブ出力がコアボード40側にP C I バス経由で返され、P C I 割込み信号がハイレベルに戻される、即ち、割込み信号がクリアされる。これは、P C I 割込み信号を長期間にわたってローレベルにしておくと、マザーボード30側のP C I 割込み処理が早く終わってしまった場合に、同じイベントによるP C I 割込み信号が二重受け付けされる可能性があるからである。また逆に、P C I 割込み信号がローレベルである期間が余りにも短いと、割込みが受け付けられないことになるからである。

【0106】

この後のP C I 割込み処理では、I/Oドライバ30Dによるアプリケーション処理が行われ、マザーボード30のP C I 割込み処理が完了した時点で、マザ

ーボード 3 0 は、P C I 割込みライン A の状態がハイレベル（ノンアクティブ）になっていることを確認するフェールセーフ処理を行う。これは、P C I 割込みライン A がローレベル（アクティブ）のまま P C I 割込み処理を終了すると、再度割込みが発生してしまうからである。そして、マザーボード 3 0 の P C I 割込み処理が完了した時点で、P C I 割込みラインがノンアクティブの場合は P C I 割込み処理を終了し、P C I 割込みラインがアクティブの場合は P C I 割込み信号を再度クリアしてから P C I 割込み処理を終了する。

（2 - 3）P C I アクセス方法

図 1 3 は P C I アクセス方法を説明するものである。P C I アクセスでは基本的にマザーボード 3 0 から P C I バス 3 9 を経由してコアボード 4 0 の F P G A 4 2 のレジスタに直接アクセスしてデータの読み出し又は書き込みを行い、データを取得したり設定したりする。一方、マザーボード 3 0 とコアボード 4 0 との間で大量のデータを P C I バス 3 9 を経由して送受信する場合は、バースト転送が使用され、パケットでデータ転送が行われて通信効率を上げることが行われる。DMA 通信、A D 変換結果、等はこの方式で行う。

【0 1 0 7】

この場合、図 5 で説明した経路 C のように、一旦コアボード 4 0 側にデータが取り込まれ、コアボード 4 0 側の I / O ドライバ 4 0 D を経由してデータの送受信が行われる。即ち、F P G A 4 2 からはデータのまとまりをマイクロコンピュータ 4 0 M に送り、マイクロコンピュータ 4 0 M 側では取り込んだ大量のデータをバースト転送でマザーボード 3 0 に送る。

【0 1 0 8】

ところで、バースト転送を使用する場合、通信相手がバースト転送対応となっている必要がある。しかしながら、F P G A 4 2 はバーストモードに対応が出来ていないため、バースト転送で F P G A 4 2 を直接アクセスすることはできない。そこで、図 1 3 に示すように、バースト転送 C 1 でマザーボード 3 0 側からコアボード 4 0 のマイクロコンピュータ 4 0 M に対して大量のデータが送られ、コアボード 4 0 のマイクロコンピュータ 4 0 M から F P G A 4 2 には非バースト転送 C 2 で少量ずつデータの設定が行われるのである。また逆に、コアボード 4 0

からデータがマザーボード 3 0 に送られる場合は、F P G A 4 2 からのデータを非バースト転送 C 2 でコアボード 4 0 のマイクロコンピュータ 4 0 M が少量ずつ取り込み、取り込みが完了すると受信割込みフラグがオンとなり、信号ライン A を通じてマザーボード 3 0 に割込みタイミングが通知される。この後、所定のタイミングでコアボード 4 0 のマイクロコンピュータ 4 0 M からバースト転送 C 1 で大量のデータがマザーボード 3 0 に送られる。

(2 - 4) 割込みコントローラの動作

図 1 4 は P C I 割込み処理を実行する割込みコントローラの動作を示すフローチャートである。ステップ 1 4 0 1 ではマザーボード 3 0 の I / O ドライバ 3 0 D による P C I 割込み信号の処理が行われる。そして、続くステップ 1 4 0 2 では割込みコントローラによって割込みフラグが取得される。この後、割込みコントローラによって取得された割込みフラグの値がステップ 1 4 0 3 で内部 R A M に記憶され、続くステップ 1 4 0 4 では取得できた割込みフラグのみがクリアされる。

【 0 1 0 9 】

なお、割込みフラグの操作中に発生した割込みフラグは、これが取得できた場合は割込み処理を行い、割込みフラグがクリアされる。また、取得出来なかった場合は割込み処理は行われず、割込みフラグはクリアされない。即ち、発生した割込みフラグに対しては、1 回のみの付随する割込み処理が実行される。この処理により、割込みフラグの取りこぼしを防止することができる。

【 0 1 1 0 】

ここで、割込みフラグの操作とは、マザーボード 3 0 で割込みフラグを F P G A 4 2 より取得する、値が 1 の割込みフラグをマザーボード 3 0 で受け付ける、割込みフラグを取得したデータをそのまま使用してクリア要求として F P G A 4 2 に書き込む、及び、マザーボード 3 0 で受け付けた割込みフラグ（値 1 の割込みフラグ）のみクリアする、の操作のことを言う。

【 0 1 1 1 】

図 1 5 に F G P A のレジスタ構成（キャプチャの場合）を示す。（a）はデータの読み書き用の割込みフラグレジスタの構成を示しており、読み出し（Read）

の場合は、1 が割込み要因が有る場合を示し、0 が割込み要因が無い場合を示している。割込み要因があるレジスタの内容が読み込まれる。また、書き込み (Write) の場合は、1 の書き込みでデータがクリアされ、0 の書き込みでデータが保存されるものとする。このように、この実施例では、割込みフラグが一括して取得され、1 つのレジスタにまとめて記憶されるので、効率的にアクセスが行われる。

【0 1 1 2】

また、(b) はキャプチャ値のレジスタの構成を示すものである。キャプチャ値のレジスタには割込みの内容が記録される。この実施例では割込みフラグの取得単位は 3 2 ビットであり、(b) に示すキャプチャ値のレジスタには 3 2 ビット幅のレジスタが使用されている。また、キャプチャの中にはフリーランタイムがあり、割込み (有効値) が入った時刻が記録され、時刻が記録されたと同時に 1 が入る。フリーランタイムは循環式である。このように、この実施例では、レジスタ構成が、フラグ (0, 1) とその内容を表すものに分かれている。フラグは 0, 1 のビット信号である。そして、値 1 のフラグが立っていたもののみをクリアすることにより、微妙なタイミングで入ってきたフラグの取りこぼしを無くす。

【0 1 1 3】

次のステップ 1 4 0 5 では割込みフラグが無くなったか否かが判定され、割込みフラグが全て無くなった場合はステップ 1 4 1 0 に進んで出力要求がポートに一括設定され、ステップ 1 4 1 1 において I/O ドライバにより P C I 割込み信号の処理が行われてこのルーチンを終了する。

【0 1 1 4】

一方、ステップ 1 4 0 5 で割込みフラグが無くなっていないことが判定された場合はステップ 1 4 0 6 に進み、入力情報の一括取得 (ポート、ラッチ) が実行される。そして、ステップ 1 4 0 7 で内部 R A M に記憶された割込みフラグに従って、割込みハンドラが起動され、複数のフラグがあった場合は、ステップ 1 4 0 8 において優先度の高い割込み処理が 1 つだけ実行される。例えば、図 1 5 (b) に示した割込みフラグレジスタでは、C H 0 と C H 3 に割込み要因があるが

、この場合は、どちらかの優先度の高い方の割込みが先に実行される。この後、ステップ 1 4 0 9 では処理が完了した割込みフラグをクリアし、ステップ 1 4 0 2 に戻る。

【0 1 1 5】

ステップ 1 4 0 2 では割込みコントローラにより割込みフラグが再度読み込まれる。この処理は、割込みフラグの処理中に、別の高い優先度の割込みフラグが発生したかどうかを見るものである。これは、最初に読み込んだ複数の割込みフラグの更新を行わずに、複数処理を全部実行し終わってから新たな割込みフラグを読み込むような処理を実行すれば、優先度の高い割込みフラグの反映が遅くなるからである。この結果、サンプリング期間が短くなり、割込みフラグのオンから処理開始までの時間が短縮される。

【0 1 1 6】

この後は、割込みフラグが残っている限りステップ 1 4 0 2 からステップ 1 4 0 9 の処理が繰り返され、割込みフラグが全て無くなった時点でステップ 1 4 0 5 からステップ 1 4 1 0 に進み、ステップ 1 4 1 0 とステップ 1 4 1 1 の処理が済むとこのルーチンが終了する。

【0 1 1 7】

図 1 6 は、サンプリング周期中に何らかの割込みイベントが発生した場合の、本発明の処理を説明するものである。ここで、「割」は割込み信号、「待」は待機状態、「処」は処理を示している。この図に示すように、最初のアプリケーション処理実行中に割込みイベント a, b が入った場合は、次のサンプリング周期の中でこの割込みイベント a, b の実行が行われ、この割込みイベント a, b の実行中に入った割込みイベント c は、次のサンプリング周期で実行される。同様に、次のサンプリング周期の中でこの割込みイベント c の実行が行われ、この割込みイベント c の実行中に入った割込みイベント d, e, f は、次の周期で実行される。

【0 1 1 8】

なお、割込みフラグはあるタイミングでまとめて見ているので、1 個の場合も見ると、複数の場合はまとめて取得した後に見る。

【0 1 1 9】

図 1 7 は、割込みイベントが途切れた場合の本発明の処理を説明するものである。図 1 6 同様に、「割」は割込み信号、「待」は待機状態、「処」は処理を示している。この図に示すように、最初のアプリケーション処理実行中に割込みイベント a, b が入った場合は、次のサンプリング周期の中でこの割込みイベント a, b の実行が行われる。この後、割込みイベントの発生がしばらく無かった場合は、コアボードは割込みイベント発生待ちになり、割込みイベント c の発生により、コアボードとマザーボードの通信が再開され、割込みイベント c が入ったら、次のサンプリング周期の中でこの割込みイベント c の実行が行われ、この割込みイベント c が実行される。

(2-5) 複数のコアボードの同期処理

コアボード 4 0 は、F P G A に容量の制限があるので、複数になるケースがあり、その時はマザーボード 3 0 が複数の F P G A に対して同期処理を行う。ここでは、コアボード 4 0 が 2 つある場合の同期処理について、図 1 8 を用いて説明する。図 1 8 では、1 枚目のコアボード 4 0 がマスタ・タイマ系のコアボードであり、2 枚目のコアボード 4 0 がスレーブである。

【0 1 2 0】

マスタ・タイマ系のコアボード 4 0 にはキャプチャ、コンペア、PWM等のいろいろな割込み機能があり、スレーブのコアボード 4 0 には、U A R T / D M A、U A R T、P W M、A D 等の通信系の割込み機能がある。マスタ・タイマ系のコアボード 4 0 には、フリーランタイマ 6 9 があり、このタイマ 6 9 の基準で動作するコンペア、キャプチャ機能は、フリーランタイマ 6 9 と同期をとる必要がある。このタイマ同期を容易にするために、タイマ系リソース（フリーランタイマ 6 9、コンペア、キャプチャ）を 1 つのコアボード 4 0 にまとめて配置し、フリーランタイマ 6 9 に依存しないリソース（通信、P W M、A D、ポート）はスレーブのコアボード 4 0 に配置している。

【0 1 2 1】

この実施例では、処理が実行される順番が括弧付き数字で示されている。

(1) では、割込みフラグの集約が行われる。O R は割込みフラグのどれかが立

っていた（オンになっていた）場合を示す。割込みフラグが立っていた場合は、P C I 割込みを発生させる。即ち、マスタ・タイマ系のコアボード 4 0 のキャプチャ、コンペア、P W M、或いはスレーブのコアボード 4 0 の何れかに割込みフラグが立っていた場合に、P C I 割込みを発生させる。スレーブのコアボードには、U A R T / D M A、U A R T、P W M、又は A D があり、スレーブの割込みフラグがどれか立っている場合は、それを集約してこれがマスタ・タイマ系のコアボードの拡張割込みフラグ内にセットされる。拡張割込みフラグ内には割込みフラグがスレーブ側にあるか否かを示す集約した結果フラグが 1 つだけ入る。

（２）では P C I 割込み信号が出てマザーボード 3 0 に入力される。この P C I 割込み信号により、同時に P C I 割込み禁止状態となる。

（３）ではマザーボード 3 0 において P C I の割込み処理が行われる。マザーボード 3 0 は P C I 割込みを 2 回重複してかけることはできないので、マザーボード 3 0 の処理完了するまでは、コアボード 4 0 は P C I 割込み禁止状態になる。

（４）マザーボード 3 0 における P C I 割込み処理が完了すると、P C I 割込み完了信号がマザーボード 3 0 からコアボード 4 0 に出力され、P C I 割込み禁止状態が解除される。

（５）P C I 割込み禁止状態の解除により、（１）の処理が再開される。

【 0 1 2 2 】

以上のような手順で複数の F P G A に対する同期処理が行われる。

【 0 1 2 3 】

【発明の効果】

以上説明したように、本発明のマイクロコンピュータのロジック開発装置によれば、ロジックの開発に伴う課題が解消され、マザーボードとコアボード間における I / O 情報の確実な通信を実現できると共に、I / O 情報通信のスピードを向上させること、及び、マザーボードとコアボードの演算処理能力の向上を図ることができて、より高性能のエンジン制御システムに対応することが可能なマイクロコンピュータのロジック開発装置を提供することができるという効果がある。また、本発明のマイクロコンピュータのロジック開発装置は、マイクロコンピュータの開発に当たって繰り返し利用することが可能であるので、開発コストを

低減することができるという効果もある。

【図面の簡単な説明】

【図 1】

本発明の背景となる電子制御式エンジンの制御システムにおける E C U の構成を示すシステム構成図である。

【図 2】

図 1 の E C U のロジックを開発する場合における本発明のマイクロコンピュータのロジック開発装置の全体構成を示す説明図である。

【図 3】

本発明のマイクロコンピュータのロジック開発装置のシステム構成を従来の電子機器制御用の E C U の構成と比較して示すブロック構成図である。

【図 4】

図 3 のマザーボードとコアボードの内部構成の一実施例を示すブロック構成図である。

【図 5】

本発明のマイクロコンピュータのロジック開発装置における、マザーボードとコアボードのソフトウェア構成、及び I F ボードのハードウェア構成を示す構成図である。

【図 6】

図 5 の P C I バスとコアボード内のマイクロコンピュータの実際の構成を説明するブロック構成図である。

【図 7】

マザーボードとコアボードとの間の P C I バスを通じた割込み信号の通知とそれに伴うデータの送受信を、時間の経過と共に示すタイミングチャートである。

【図 8】

図 7 に示したマザーボードにおける入力情報の取得、アプリケーション処理、及び出力要求の確定の詳細を示すソフトウェア構成図である。

【図 9】

図 7 に丸付き数字 1 で示すパワーオンシーケンスを実行するための、マザーボ

ード、コアボード、及び機能ボードの構成、及びその周辺回路の構成を示す説明図である。

【図 10】

図 9 のように構成されたマザーボード、コアボード、及び機能ボードを含む回路の、パワーオンシーケンスにおける処理ステップを記載した図である。

【図 11】

図 10 の処理ステップの続きのステップ、及びパワーオフシーケンスの処理ステップを記載した図である。

【図 12】

図 7 の割込み信号ライン A によってもたらされる P C I 割込み信号によって発生する P C I 割込み処理の詳細を示すタイミングチャートである。

【図 13】

マザーボードから P C I バスを経由してコアボードの F P G A のレジスタに直接アクセスしてデータの読み出し又は書き込みを行う、本発明における P C I アクセス方法の実施例を説明するタイミングチャートである。

【図 14】

P C I 割込み処理を実行する割込みコントローラの動作を示すフローチャートである。

【図 15】

キャプチャの場合の F P G A のレジスタ構成を示すものであり、(a) は割込みフラグレジスタの構成を示す図、(b) はキャプチャ値のレジスタの構成を示す図である。

【図 16】

サンプリング周期中に何らかの割込みイベントが発生した場合の、本発明の処理を説明するタイミングチャートである。

【図 17】

割込みイベントが途切れた場合の、本発明の処理を説明するタイミングチャートである。

【図 18】

コアボードが複数枚ある場合の、マザーボードの複数のコアボードに対して行う同期処理を説明する図である。

【符号の説明】

- 1…E C U
- 2…組込み用マイクロコンピュータ
- 7…内部タイマ
- 8…割り込みコントローラ
- 9…メモリ
- 1 0…C P U
- 1 1…内部バス
- 2 0…本発明のロジック開発装置
- 2 7…ホストコンピュータ
- 3 0…マザーボード
- 3 0 D… I / O ドライバ
- 3 1…ソフトウェアエア（エンジン制御アプリケーション）
- 3 2…P C I 通信ソフトウェア
- 3 3…C P U
- 3 4…仮想割り込みコントローラ
- 3 5…内部タイマ
- 3 6…P C I バスインタフェース
- 3 7…内部バス
- 3 8…仮想 I / O レジスタ
- 3 9…P C U バス
- 4 0…コアボード
- 4 0 D… I / O ドライバ
- 4 1…P C I 通信ソフトウェア
- 4 2…疑似マイクロコンピュータ装置（F P G A）
- 4 3…内部バス
- 4 4…P C I バスインタフェース

4 5 … C P U

5 0 … I F ボード

5 5 … 機能ボード

5 6 … 内部バス

6 3 , 6 4 … 比較器

6 5 … 電源 I C

6 7 … マイクロコンピュータ

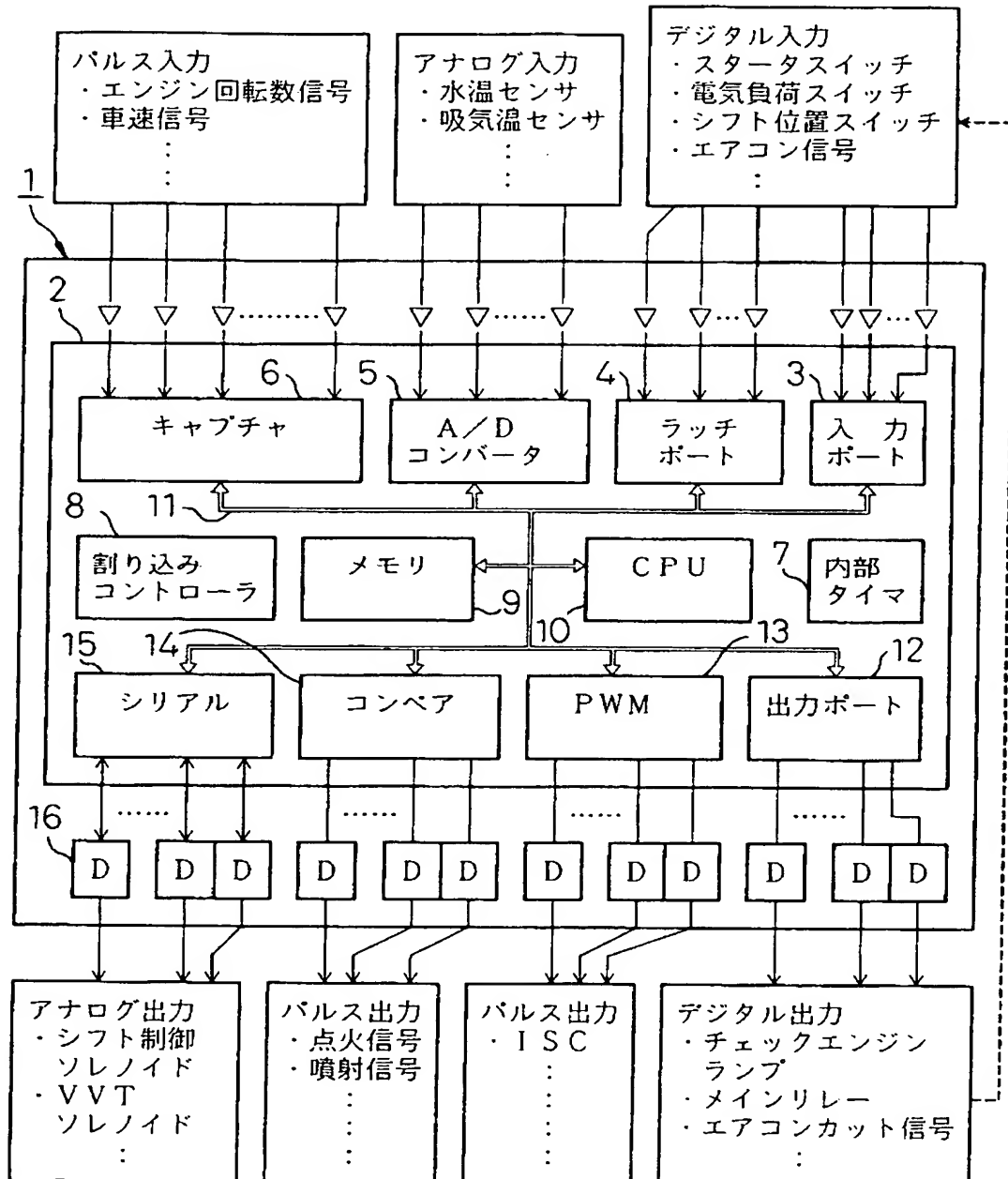
6 9 … フリーランタイマ

【書類名】

図面

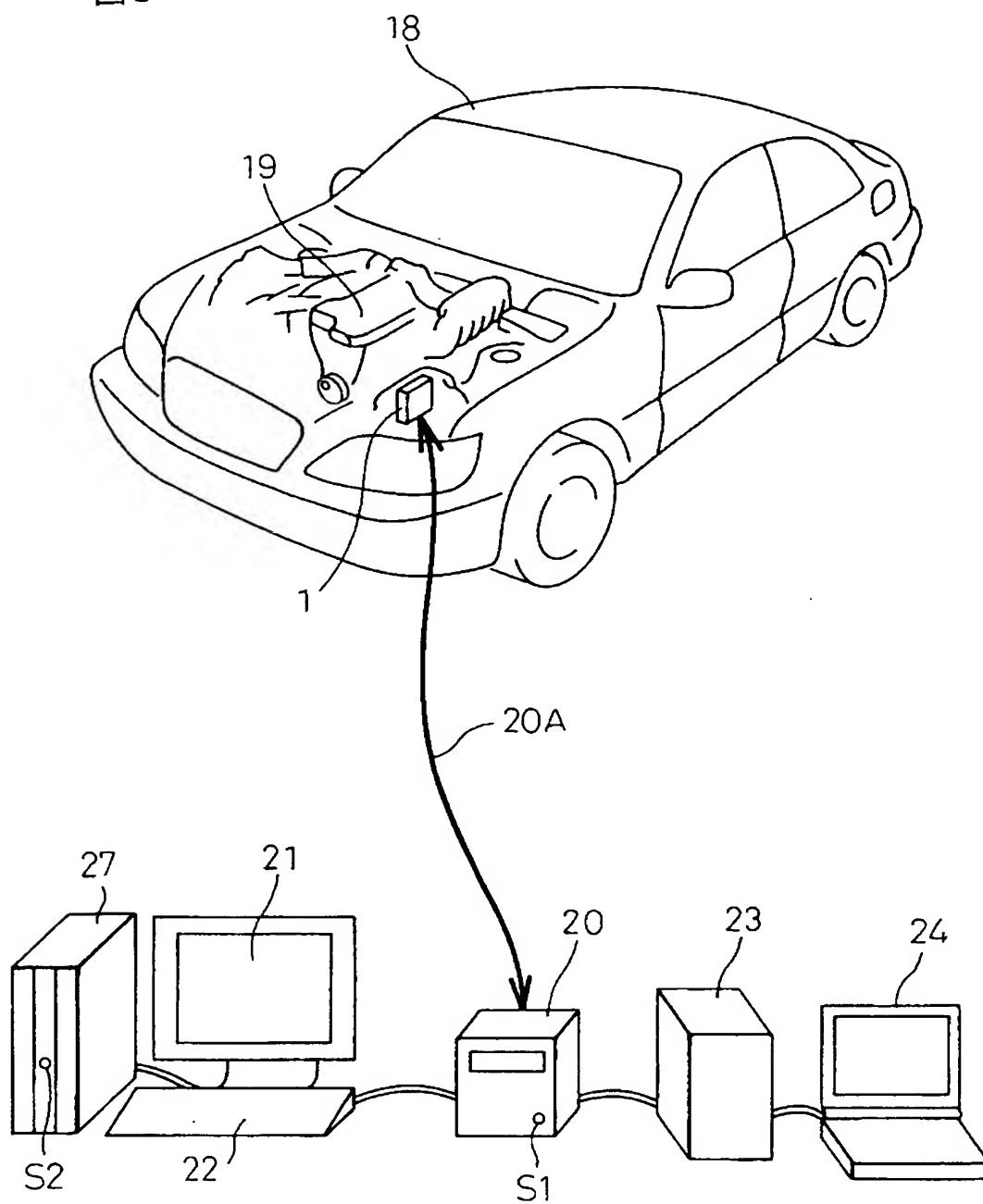
【図 1】

図 1

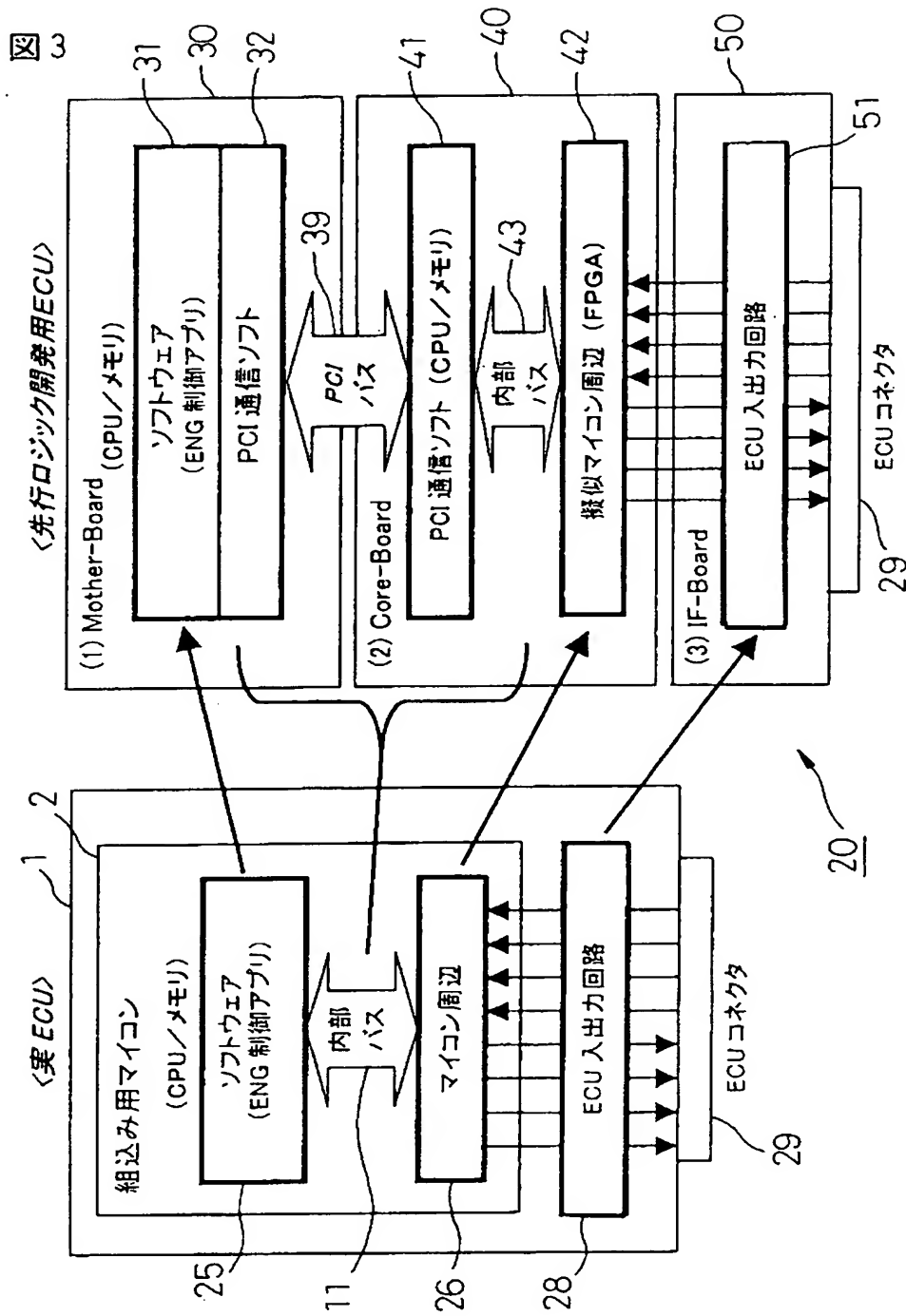


【図 2】

図 2

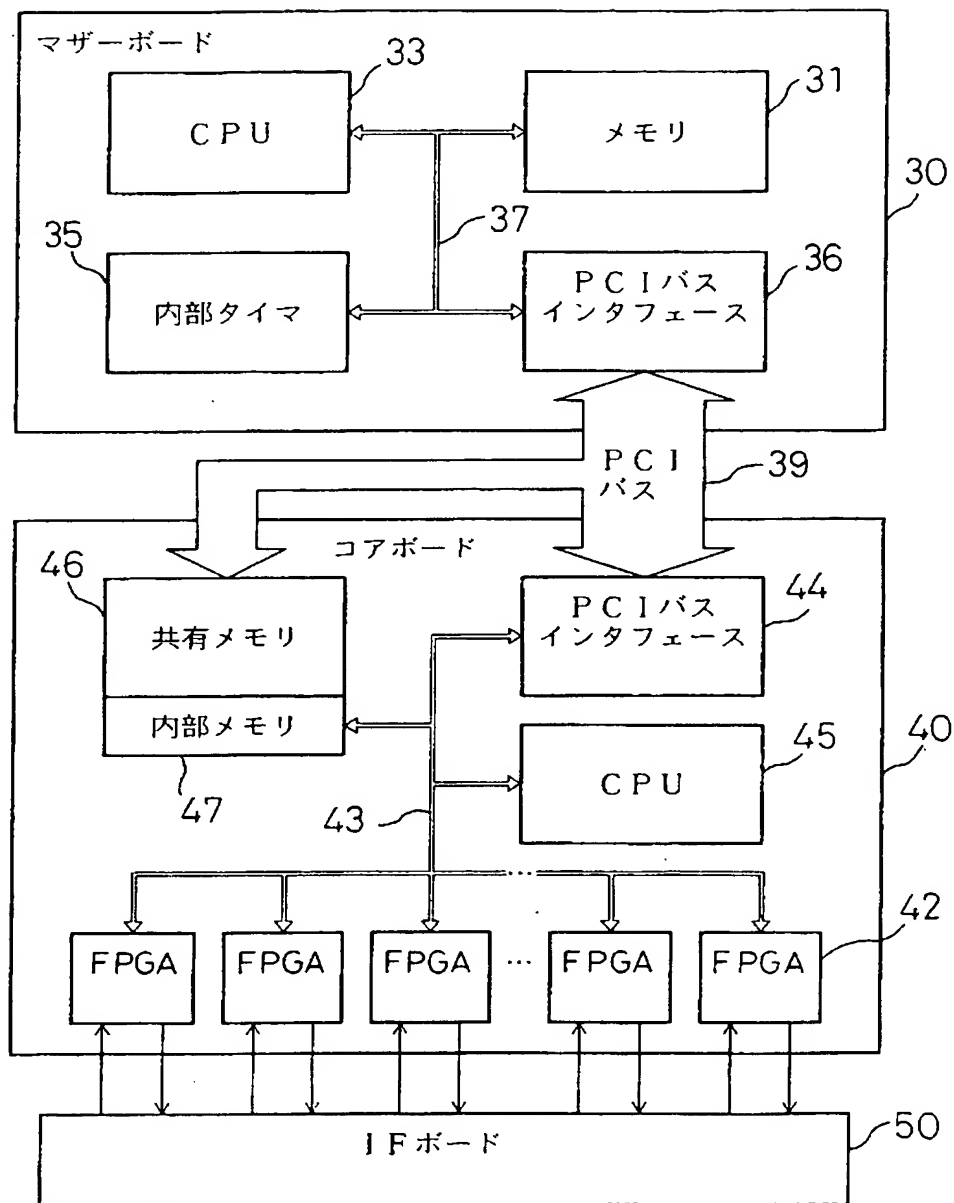


【図3】

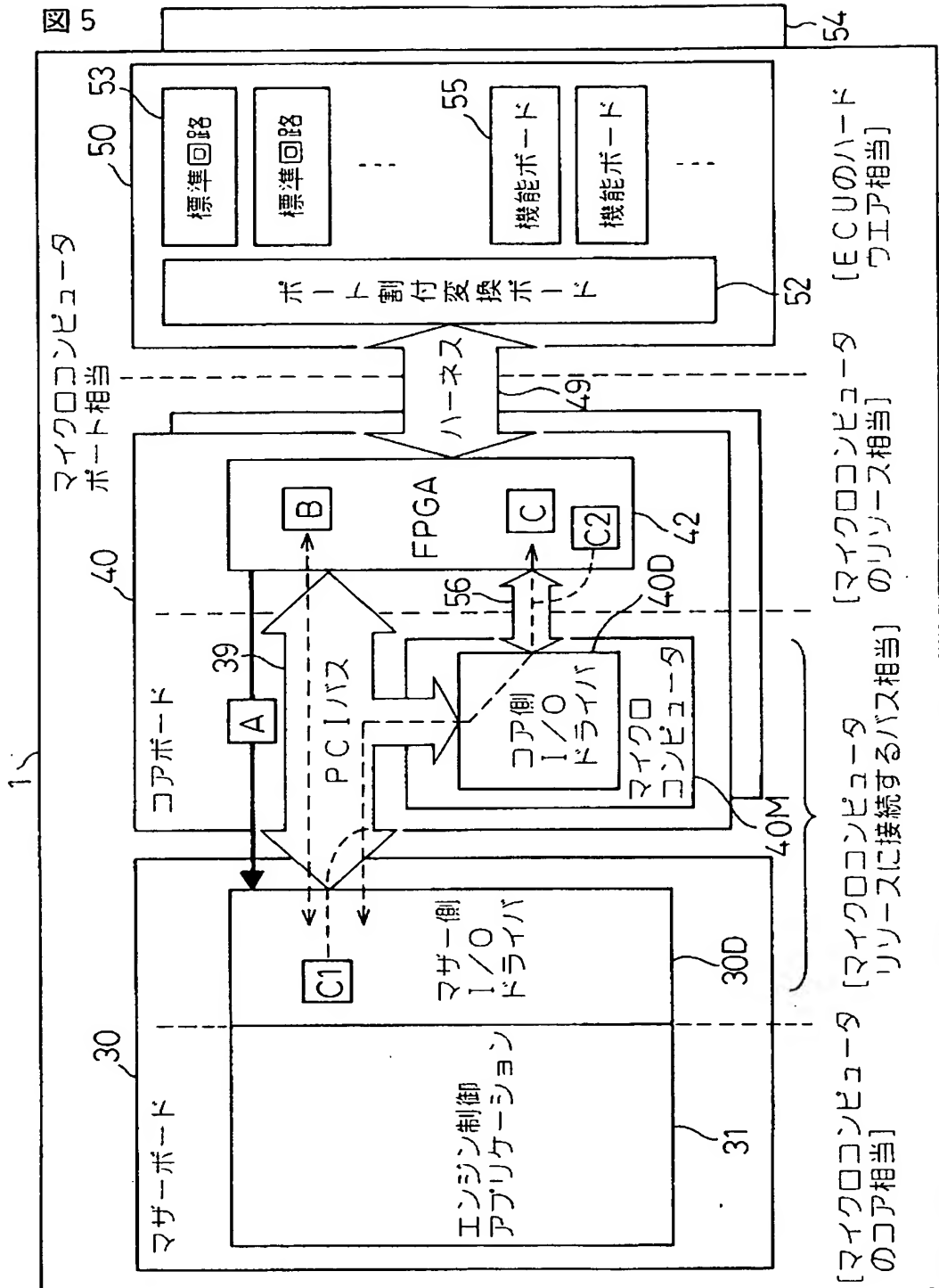


【図 4】

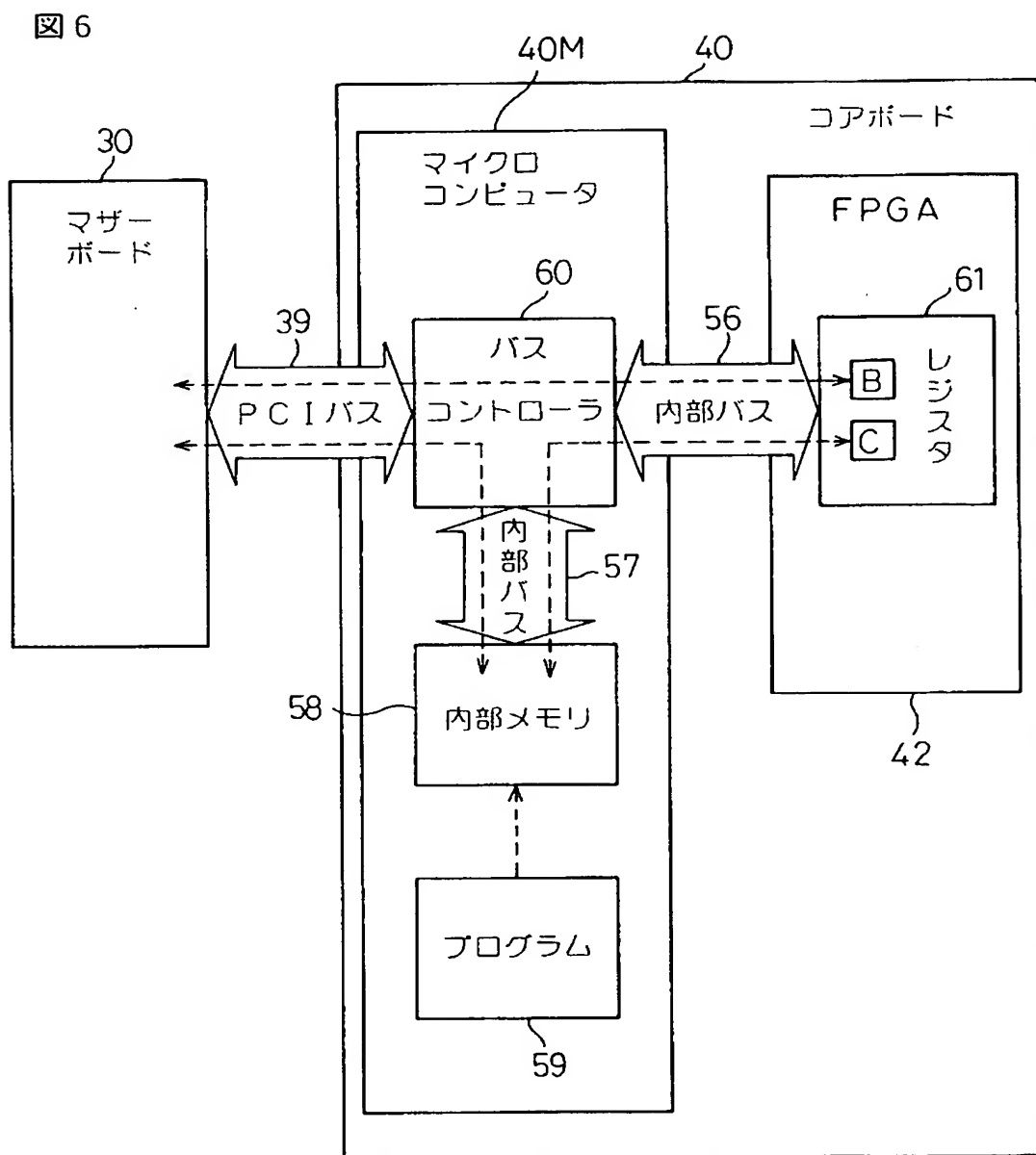
図 4



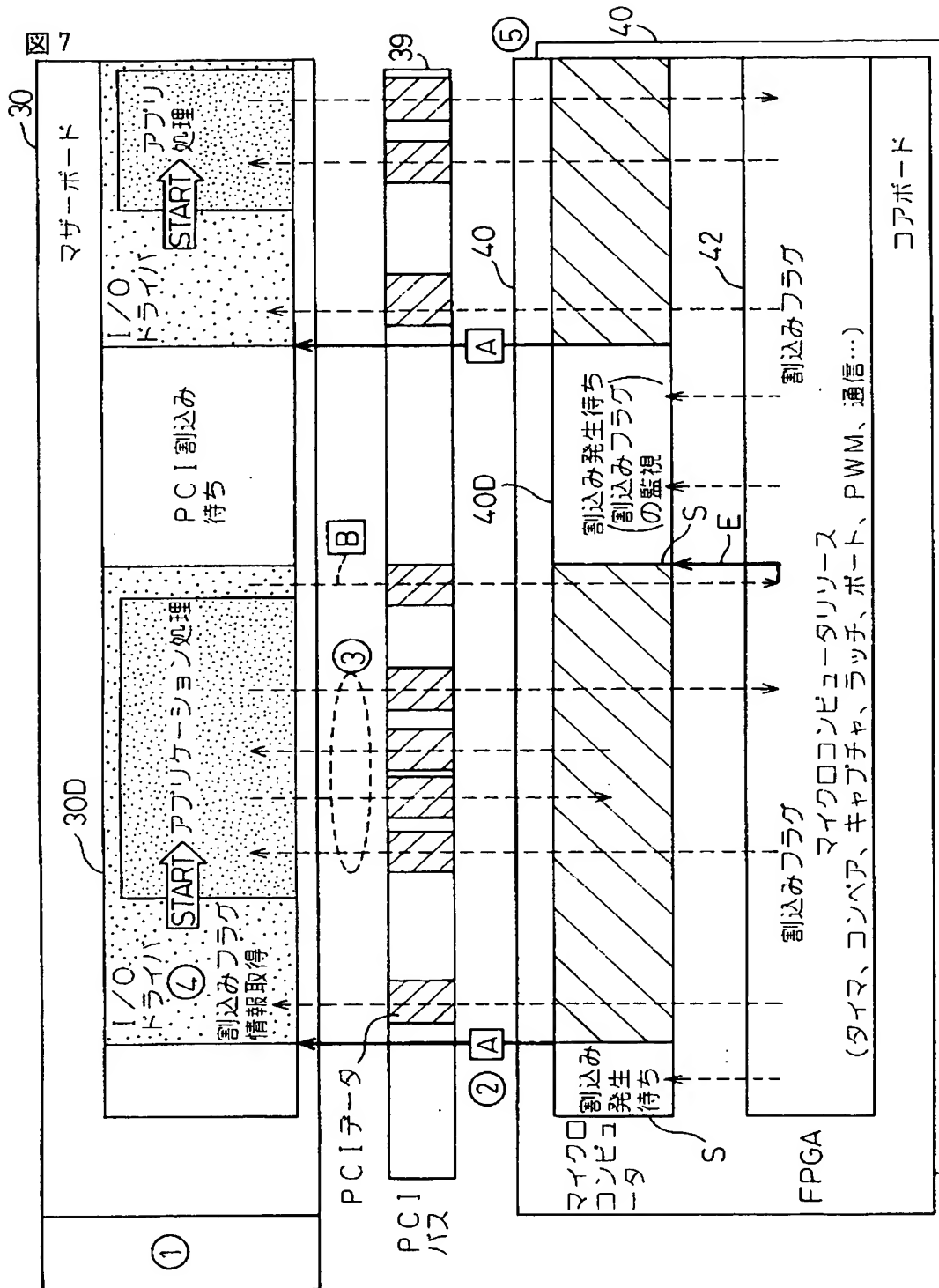
【図 5】



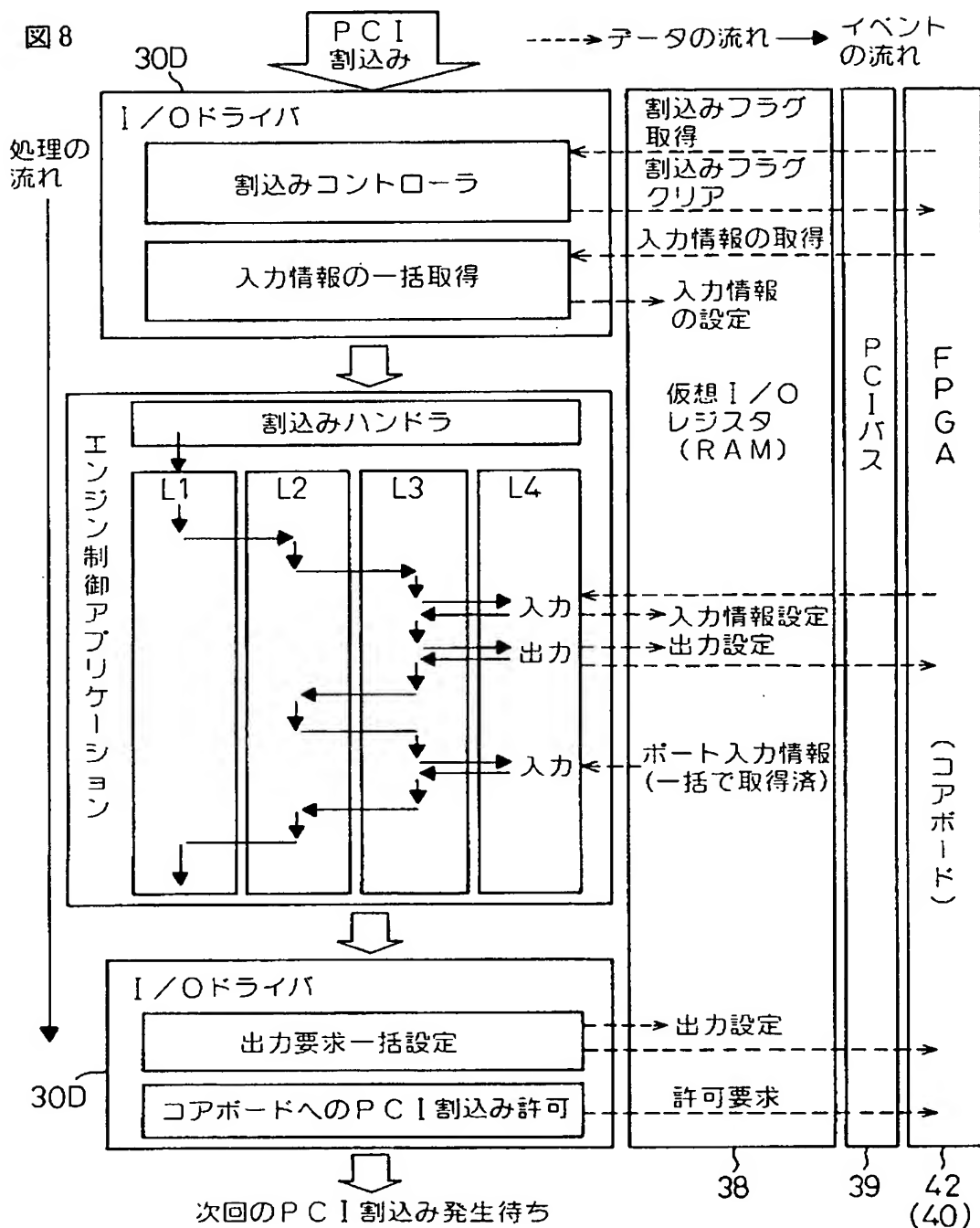
【図 6】



【図 7】

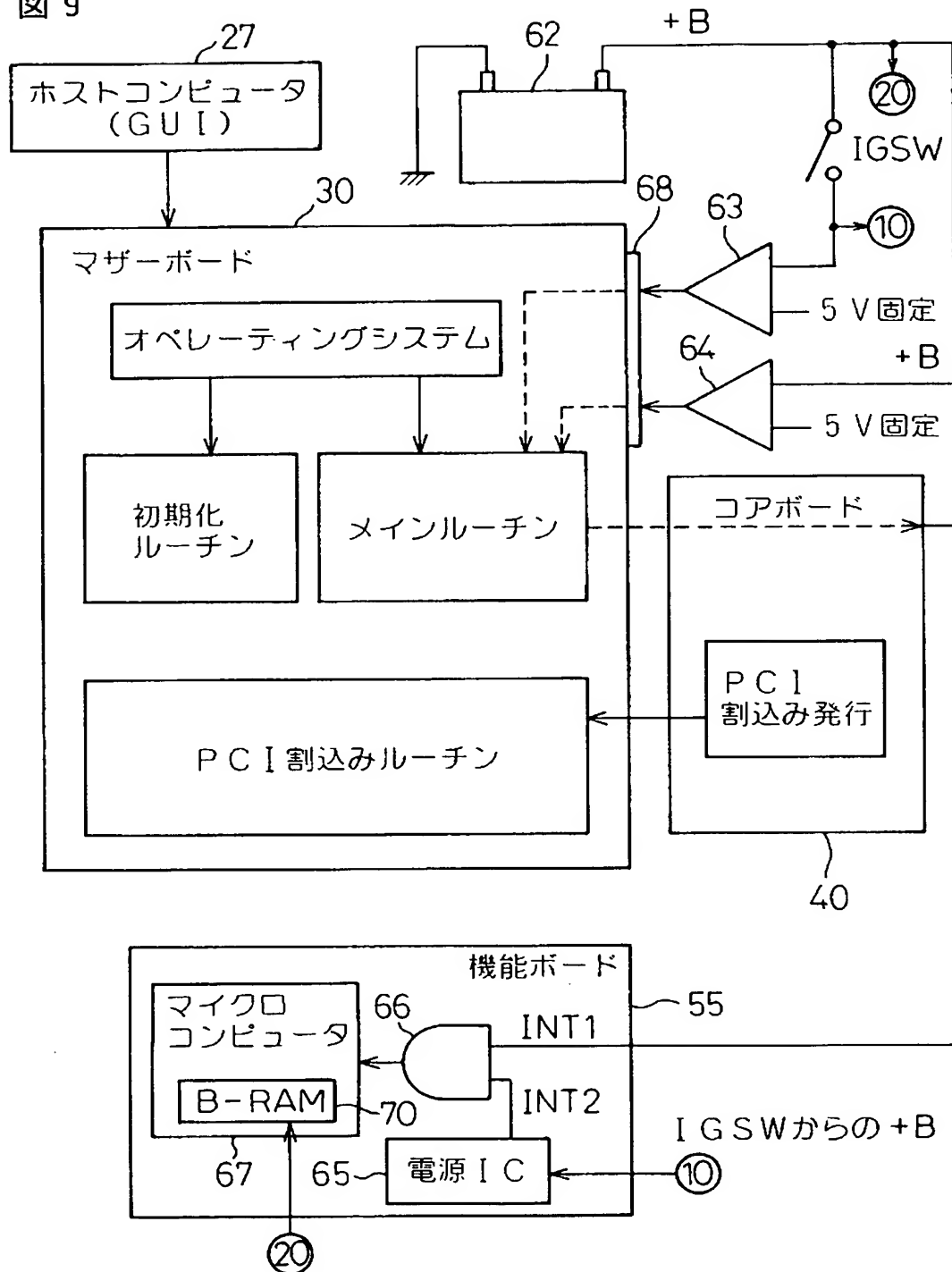


【図 8】



【図 9】

図 9



【図 10】

図 10

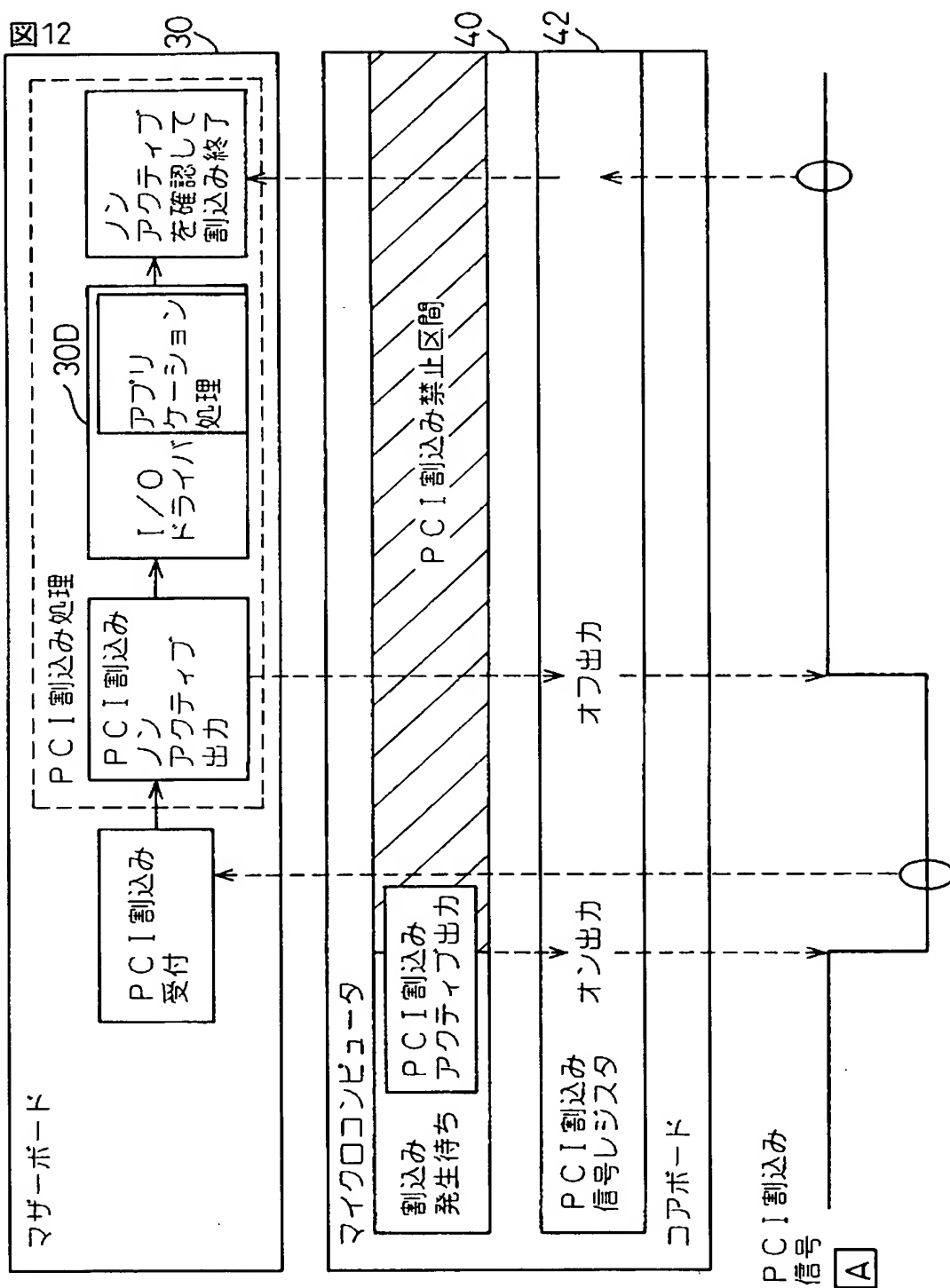
	マザーボード	コアボード	機能ボード
1	作業者のシステム電源オンでオンする	同左	電源オフ状態
2	ホストコンピュータからのソフトウェアのダウンロードにより起動する	同左 P C I 割込みは ロック状態	
3	ボートの初期値を設定する(※1) (1)コンペアをイミディエート出力で設定 (2) P W M を 0 % 出力で設定 (3) I N I T 1 を L O に設定 (4) ステート 1 を設定		
4	ステート 1 の処理 (1) I G S W のオン待ち		
5	(2) I G S W のオンの検出 (3) ステート 2 を設定		
6	ステート 2 の処理 (1) メモリ復元 (バッテリ外れ時は初期化) (2) E E P R O M データの復元 (3) ステート 3 を設定		電源が オンする
7	ステート 3 の処理 (1) パワーオンリセット待ち		パワーオン リセット時間 (電源 I C)
8	(2) パワーオンリセット時間経過		I N I T 2 が H I
9	(3) I N I T 1 を H I に設定 (4) 通信以外の初期化处理 (5) ステート 4 を設定		I N I T 1 が H I → I N T I , I N T I 2 共に H I → リセット解除
10	ステート 4 の処理 (1) 機能ボード初期化处理時間待ち		初期化处理
11	(2) 機能ボード初期化处理時間経過 (3) 機能ボードとの初期化通信起動 (4) ステート 5 を設定	DMA 通信	メイン処理 受信
12			

【図 11】

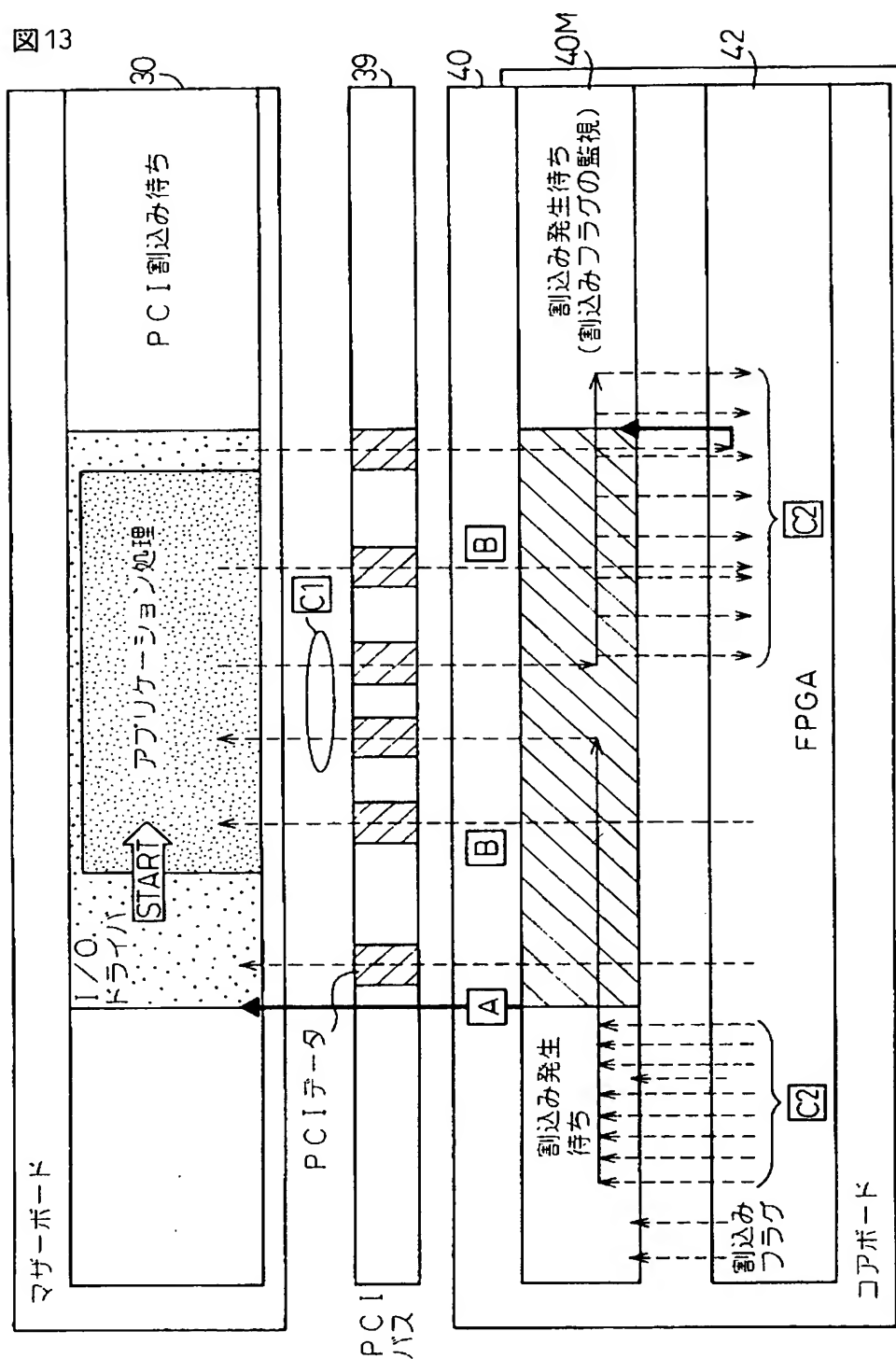
図 11

	マザーボード	コアボード	機能ボード
12	状態 5 の処理 (1) 機能ボードとの初期化通信完了待ち		受信処理 送信データ 作成
13	(2) 完了検知 ← ※完了せずにタイムアウトした 時は状態 4 に戻る		送信
14	状態 6 の処理 (1) 初期化完了		
15	(2) コアボードのロック解除要求 → (3) 状態 7 を設定	割り込みフラグ を全クリア PCI 割り込み ロック解除	
16	状態 7 の処理 (1) I G S W オフ待ち	PCI 割り込み処理	
17	(2) I G S W オフ検出 (3) 状態 8 を設定		電源オフ
18	状態 8 の処理 (1) ロック要求 →	PCI 割り込み ロック	
19	(2) メモリ、EEPROM のデータを保存	PCI 割り込み禁止	
20	(3) ボートの初期値設定 (※ 1) と同じ (処理完了後状態 1 を設定)		

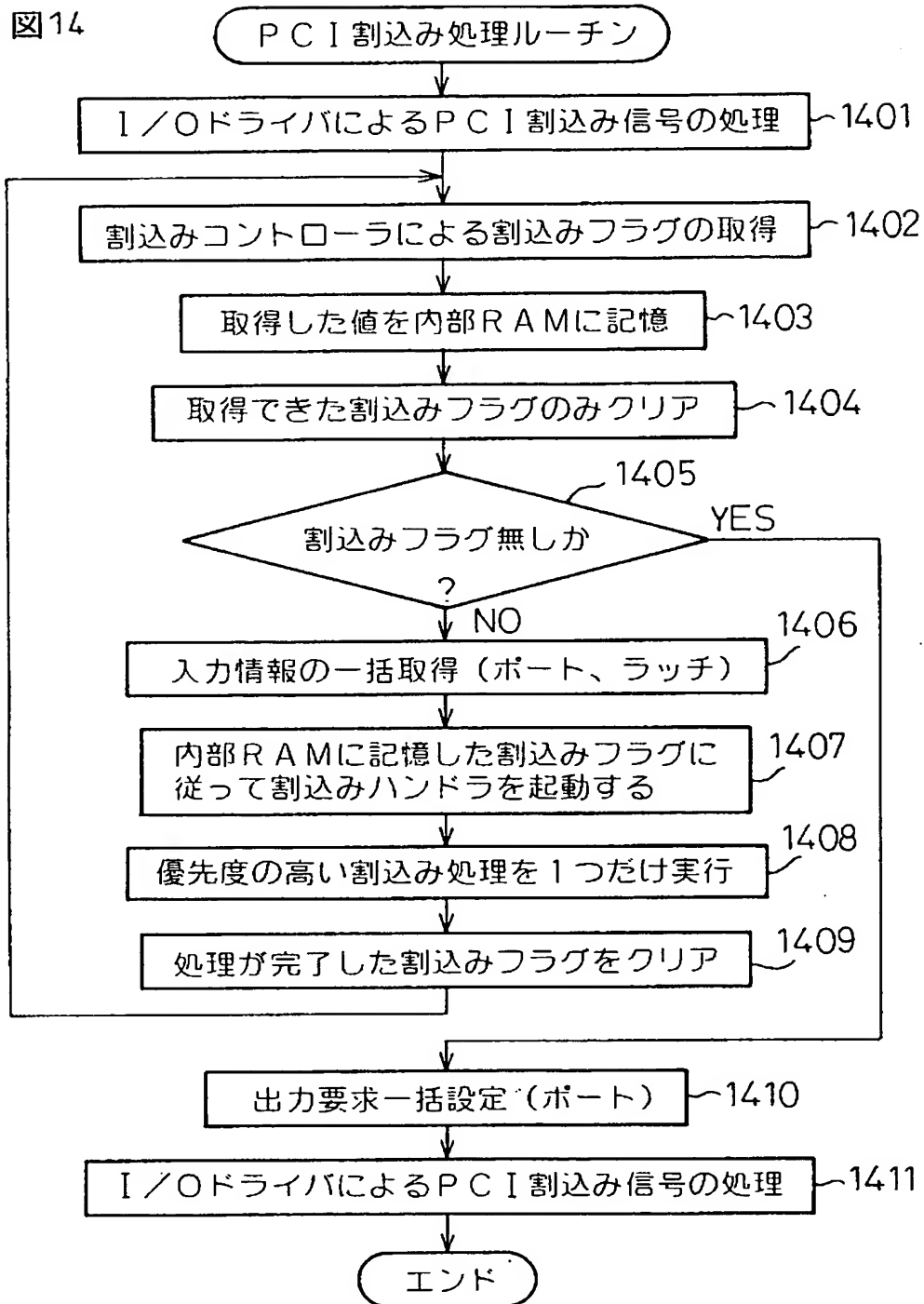
【図 12】



【図 13】



【図 14】



【図 15】

図 15

FPGA のレジスタ構成 (キャプチャの場合)

(a)

割り込みフラグレジスタ (R/W) の構成

.....	CH5	CH4	CH3	CH2	CH1	CH0
.....	0	0	1	0	0	1

Read → 1 : 割り込み要因有り 0 : 割り込み要因無し
 write → 1 書き込みでクリア 0 書き込みで保存

(b)

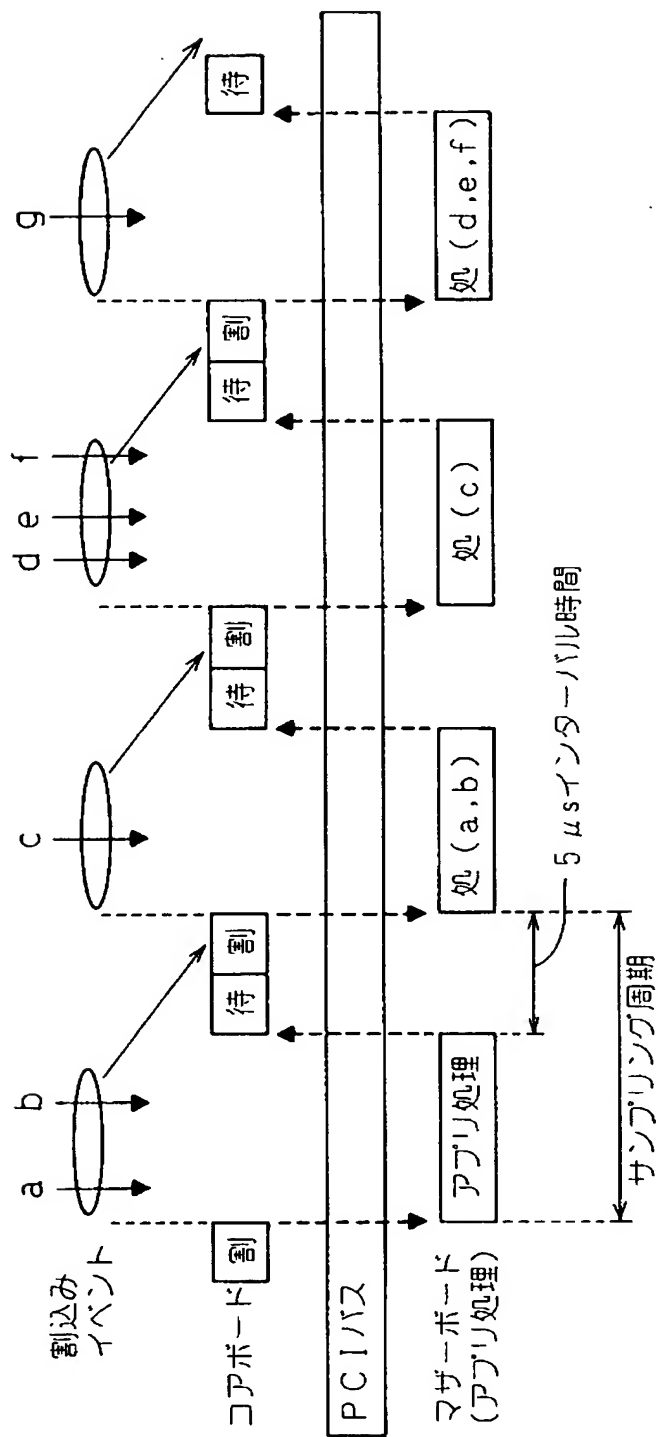
キャプチャ値のレジスタ (R)

CH0 のキャプチャ値	
CH1 のキャプチャ値	
CH2 のキャプチャ値	
CH3 のキャプチャ値	
CH4 のキャプチャ値	
CH5 のキャプチャ値	
⋮	
32ビット幅	

【図 16】

図 16

[サンプリング区間に何らかの割り込みイベントが発生]



【書類名】 要約書

【要約】

【課題】 先行ロジックに必要なCPU機能を確保でき、I/O情報の確実な通信とCPUの演算処理能力の向上を図るロジック開発装置を提供する。

【解決手段】 ECU1で使用される組込み用マイクロコンピュータ2のロジック開発装置20を、アプリケーション処理機能31と通信機能30Dを備えたマザーボード30と、疑似マイクロコンピュータ周辺装置42、演算機能40M、及び通信機能40Dを備え、マザーボード30とPCIバス39で接続されるコアボード40と、ECU1のハードウェア相当回路53, 55を有し、コアボード40に接続するIFボード50とから構成し、マザーボード30の通信機能30Dと、コアボード40の疑似マイクロコンピュータ周辺装置42との間に設けたPCIバス39により、両者間で直接データの送受信を行うようにしたものである。

【選択図】 図5

特願 2 0 0 3 - 0 2 4 7 3 3

出 願 人 履 歴 情 報

識別番号

[0 0 0 2 3 7 5 9 2]

1. 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

兵庫県神戸市兵庫区御所通 1 丁目 2 番 2 8 号

氏 名

富士通テン株式会社